LA-UR-19-22080

Approved for public release; distribution is unlimited.

Title:	NEK5000 Assessment Milestone Report: Single-Phase Natural Circulation using Direct Numerical Simulation (DNS) & Large Eddy Simulation (LES) Methods
Author(s):	Naranjo, Mario Reyes Kim, Seung Jun
Intended for:	Report
Issued:	2019-04-08 (rev.1)

Disclaimer:
Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy under contract 89233218CNA000001. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

NEK5000 Assessment Milestone Report: Single-Phase Natural Circulation using Direct Numerical Simulation (DNS) & Large Eddy Simulation (LES) Methods

Mario R. Naranjo and Seung Jun Kim
Nuclear Engineering and Nonproliferation Division
Nuclear System Design and Analysis (NEN-5)
Los Alamos National Laboratory
Los Alamos, NM 87544

Executive Summary

This study evaluates the capabilities of the NEK5000 open-source code as an alternative to the commercially available software that is currently heavily relied upon. The first three case studies initiated the author with (1) the syntax and flow of the program, (2) varying initial and boundary conditions and (3) geometries of greater complexity & dimension. Case 1 studies transient conjugate heat transfer in the entry region of a pipe. Cases 2 & 3 simulate the natural convection flow in an enclosed differentially heated cavity (DHC) of two and three dimensions, respectively. The final case study, Case 4, produces three-dimensional single-phase natural convective flow using the Boussinesq approximation within a simplified representation of an aqueous homogeneous reactor system. The Boussinesq flow in Case 4 clearly depicts both Rayleigh-Taylor and Kelvin-Helmholtz instabilities qualitatively during post-processing analysis. Further analysis of Case 4, up to the point of a quasi-steady state (i.e. thermal-equilibrium condition), is done using the LES solver method in NEK5000. The volume-averaged solution temperatures, when compared between solvers, showed no significant difference. Due to its minimal compromise in fidelity of solution resolution and inexpensive computing cost, compared to the DNS solver, the LES solver is recommended for the current application. Methodology development for the four case studies are reported and a preliminary optimization study with different polynomial resolutions for Case 4 is addressed as a resource when performing future simulations using NEK5000. Hence, this report can be used as a reference starting point to the end user of NEK5000 for natural circulation phenomena in nuclear reactor safety applications.

Contents

Executive Summary 1
1 Nomenclature 5
2 Background and Introduction 6
3 NEK5000 7
4 Case Studies 9
4.1 Case 1: 2-D Pipe Flow with Uniform Heat Flux at the Wall 9
4.1.1 Problem Definition 9
4.1.2 Simulation and Analysis 12
4.2 Case 2: 2-D Natural Circulation with Flux Boundary Conditions 16
4.2.1 Problem Definition 16
4.2.2 Simulation and Analysis 19
4.3 Case 3: 3-D Natural Circulation with Flux Boundary Conditions 24
4.3.1 Problem Definition 25
4.3.2 Simulation and Analysis 26
5 Case 4: Natural Convection in an AHR 29
5.1 Problem Definition 31
5.2 Simulation and Analysis 31
5.2.1 Output File Size 31
5.2.2 Computation Time 32
5.2.3 Precision of Solution 32
6 Case 4: LES (NEK5000) 38
6.1 Problem Definition 38
6.2 Simulation and Analysis 39
6.2.1 Computation Time 40
6.2.2 Precision of Solution 41
7 Conclusion and Future Work 42
8 References 43
Appendices 47
1 Input Files 47
1.1 moly.box File 47
1.2 SIZE File 47
1.3 moly.par File 48
1.4 moly.usr File 49

1.4.1 subroutine uservp ... 49
1.4.2 subroutine userf ... 49
1.4.3 subroutine userq ... 50
1.4.4 subroutine userbc ... 50
1.4.5 subroutine useric ... 50

Figures

Figure 1: 7x7 Spectral Elements (E=49) with data in terms of $7th$ order Legendre Polynomial ... 8
Figure 2: Case 1 Velocity Only ... 10
Figure 3: Case 1 Heat Flux Only ... 11
Figure 4: Cutout of Figure 5 with where initial velocity (5cm/s) is transparent and highlighted with blackline ... 12
Figure 5:Temperature & Velocity profile of Case 1 ... 12
Figure 6: Thermal profile is fully developed at 13 seconds ... 13
Figure 7: Velocity profile is fully developed at 4 second ... 13
Figure 8: Temperature Profiles ... 14
Figure 9: Velocity Profiles ... 15
Figure 10: Numerical Simulation Profiles & Analytical Boundary Layers ... 16
Figure 11: Plot of Case 2-1 Boundary Conditions ... 17
Figure 12: Plot of Case 2-2 Boundary Conditions ... 18
Figure 13: Plot of Case 2-3 Boundary Conditions ... 19
Figure 14: Case 2-1 Simulation Beginning ... 20
Figure 15: Case 2-2 Simulation Beginning ... 20
Figure 16: Case 2-3 Simulation Beginning ... 21
Figure 17: Case 2-1 Thermally Stratified Core ... 21
Figure 18: Case 2-2 Some Stratification ... 21
Figure 19:Case 2-3 Mixing Condition at 1 Minute ... 21
Figure 20: Case 2-2 Thermally Stratified Core at 4 Minutes$(Ra = 1.2 \times 109 < RaF = 2.59 \times 1012)$ *... 22*
Figure 21: Isotherms for water in the case of adiabatic horizontal walls (Ra = 3 x10^9*) (Henkes and Hoogendoorn 1990) ... 22*
Figure 22:Geometry of the Problem (Henkes and Hoogendoorn 1990) ... 23
Figure 23: Case 2-3 Mixing at End Time @ 5 min ... 24
Figure 24: Plot of Boundary Conditions ... 25
Figure 25: Case 3-1 Vertical Turbulence at 36 seconds ... 27
Figure 26: Case 3-2 Vertical Turbulence at 3 Seconds ... 28
Figure 27: Case3-1 @ 1 minute 48 seconds (12 x 9 sec) ... 28
Figure 28: Case3-2 @ 9 seconds ... 28
Figure 29: Mesh in the xy-plane ... 30
Figure 30: 2-D Enlarged View of xz-plane ... 30
Figure 31: 2-D View of xz-plane ... 30
Figure 32: Rayleigh-Taylor Instabilities along Walls ... 33
Figure 33: Termination Point of N=6 @ t = 127 ... 33
Figure 34: Rayleigh-Taylor&Kelvin-Helmholtz Instabilities @ t = 134 ... 33
Figure 35: Fully Developed Kelvin-Helmholtz Instabilities @ t = 136 ... 34
Figure 36: Prescribed End Time @ t = 180 ... 34
Figure 37: RTI evolution @ t = 127 seconds ... 35
Figure 38: KHI Develop @ t = 136 seconds ... 35
Figure 39: Velocity in the z-direction at t = 136 seconds ... 36
Figure 40: Vertical Velocity Components at t=136 seconds ... 36

Figure 41: Volumetric Average Temperature Comparison 37
Figure 42: Comparison of Temperature Differences by Polynomial Order 38
Figure 43: moly.par LES Settings 38
Figure 44: moly.par Restart Setting 39
Figure 45: Temperature Difference (DNS - LES) Prior to Instabilities 39
Figure 46: DNS-LES RMS Polynomial Order Comparison 40
Figure 47: LES Volumetric Average Temperature Comparison 41
Figure 48: Geometery File 'moly.box' 47
Figure 49: SIZE File 48
Figure 50: Parameter File 49
Figure 51: uservp Subroutine 49
Figure 52: userf Subroutine 49
Figure 53: userq Subroutine 50
Figure 54: userbc Subroutine 50
Figure 55: useric Subroutine 50

Tables

Table 1: Water Properties at 1 bar (MatWeb, LLC. 2018) 11
Table 2: Uranyl Sulfate Properties (Kim and Buechler 2016) 31
Table 3: DNS & LES Polynomial Order's Impact on File Size 32
Table 4: DNS-Polynomial Order's Effect on Runtime 32
Table 5: LES-Polynomial Order's Effect on Runtime at t =127 Seconds Simulation Time 41

1 Nomenclature

Gr	= Grashof number
Nu	= Nusselt number
Pr	= Prandtl number
Ra	= Rayleigh number
$\mathrm{Ra_F}$	= Flux Rayleigh number
Re	= Reynolds number
$\mathrm{c_p}$	= specific heat
$\boldsymbol{f}$	= body-force term
$\boldsymbol{g}$	= gravitational acceleration
L	= characteristic length
n	= Normal vector
q'' or f	= flux
$\mathrm{q_{vol}}$	= dissipation function
r	= heat source per unit mass
T	= temperature
T_0	= initial temperature
T_s	= temperature at surface
T_∞	= bulk temperature
α	= thermal diffusivity
β	= volumetric thermal expansion coefficient
κ	= coefficient of thermal conductivity
μ	= dynamic viscosity
ν	= kinematic viscosity
ρ	= density
ρ_0	= density at T_0

2 Background and Introduction

In the United States there are actively 31 private nuclear research and test reactors, in addition to several Department of Energy facilities (United States Nuclear Regulatory Commission 2018). Due to the limited supply of research facilities compounded with stringent safety and regulatory protocols, experimentation on current and emerging nuclear technologies is logistically intensive and monetarily expensive. Current advancements in computing technology provide a great opportunity for the next generation of computational fluid dynamics (CFD) using spectral methods. Natural convection flow inside a differentially heated cavity is “a popular problem to test and compare numerical algorithms that are developed to solve the Navier-Stokes equations” (Janssen and Henkes 1995). Natural circulation is a key safety feature in Aqueous Homogeneous Reactors (AHRs). Conceptualized at Los Alamos National Laboratory (LANL), initial experiments using AHRs were completed at LANL and Oak Ridge National Laboratory (ORNL). The reactor design was discovered to be an efficient means of production for medical radioisotopes (e.g. molybdenum-99 and other fission products) due to its streamlined process of extracting target radioactive sources from the fissile solution via a separation system. As the demand for medical radioisotopes has increased so has the need for a safe and low-cost isotope source, the most commonly used of which is technetium-99m (^{99m}Tc), daughter product from molybdenum decay. Studies indicate solution reactors pose a significantly lower hazard to the workers, public and environment, compared to other modern reactors (International Atomic Energy Agency 2008). AHRs inherent safety features (1) passively maintain a low solution temperature, below boiling (90℃), by natural convection cooling, (2) self-regulate reactivity control by the negative reactivity feedback effect and (3) low reactor vessel pressure (less than atmospheric). Contrary to the majority of medical isotope production reactors (MIPRs) that irradiate targets in a neutron flux typically generated by reactor fuel, the fuel solution is the target for AHRs. Solution reactors produce an equivalent quantity of ^{99}Mo, while reducing the power consumption and generation of radioactive waste by a factor of approximately 100. The inherent safety, efficient use of radioactive fuel and ability to selectively extract radionuclides from its fuel solution make AHRs particularly well suited for medical isotope production (Mario R. Naranjo 2018).

We use the open-source Spectral Element Method (SEM) code Nek5000’s direct numerical simulation (DNS) and Large Eddy Simulation (LES) models to simulate natural circulation in an AHR as unsteady incompressible flow with active scalar transport, e.g. temperature. Temperature is considered an active scalar due to the buoyancy term’s impact on velocity via the Boussinesq approximation. Among the first used in practical flow simulations due to their high-order of accuracy, spectral element schemes allow us to attain an accurate solution with a lower number of grid points (Schlatter 2010). “The schemes are “infinite”-order accurate if the expansion functions are properly chosen…for the same number of points, the spectral element method is several orders-of-magnitude more accurate than the low-order finite element technique (Patera 1984). Modern advancements in High Performance Computing (HPC) enable the use of SEM solvers to evaluate high-fidelity thermal hydraulic simulations as computational resources are increasingly accessible and affordable. Reusing a reasonable mesh geometry, exponential convergence to the exact solution is achieved by increasing the polynomial. This study of numerically simulating the thermal hydraulic phenomenon of natural circulation occurring in AHRs is valuable on multiple fronts. First, it serves as an introduction to a high-order numerical scheme with no license costs, i.e. open-source. Second, it presents itself as a resource for future benchmark studies of, the license cost heavy, traditional Finite Volume Method (FVM) codes. Lastly, this allows AHR designers an option when optimizing nuclear resource use to meet increasing demand for medical isotope production reactors (MIPRs).

3 NEK5000

Created at Argonne National Laboratory's Mathematics and Computer Science Division, Nek5000 is a SEM time-stepping code written in C and Fortran 77. Nek5000 is a highly scalable variant of Nekton 2.0, both the first three-dimensional SEM and commercially available code for parallel computing (Argonne National Laboratory 2018). It was recognized with the Association for Computing Machinery's (ACM) Gordon Bell Prize in 1999. The program uses DNS or LES models to simulate unsteady incompressible flow with thermal and passive scalar transport. NEK5000 can also be used to simulate compressible flow of low Mach number.

Spectral methods involve the expansion of the solution to a differential equation in a high-order orthogonal expansion, the coefficients of which are determined by a weighted-residual projection technique (Patera 1984). Due to the spectral method's high order of accuracy using a lower number of grid points, it was amongst the first used for practical flow simulations (Schlatter 2010). Originally developed by meteorologists, the spectral method was first proposed and numerical computations conducted in the 1940's & 50's. The computational expense of directly calculating nonlinearities limited the usage of spectral methods until the 1970's (Hussaini and Zang 1986). Since the 70's, technological and spectral method technique advancements have made its usage more reasonable and efficient.

NEK5000 accomplishes spatial discretization using the method of weighted residuals with Gauss-Legendre quadrature, allowing the freedom to choose both node locations, x_i, and weighting coefficients, w_i. Each element's solution and data are expanded in terms of Gauss-Lobatto-Legendre (GLL) collocation points, whose associated weights are based on an $((\mathrm{N}-1)\text{ or }n)^{\mathrm{th}}$ order Legendre polynomial, P_n. Refining within each spectral element an n^2 or n^3 mesh, for two or three dimensions, respectively. Figure 1 illustrates 7x7 spectral elements each containing a tensor-product grid of GLL nodes for 7^{th} order Legendre polynomial space. Gauss-Legendre quadrature primarily differs from Newton-Cotes quadrature in that the spacing of the nodes are not equidistant. The Legendre polynomials are part of a family of orthogonal polynomials by which all data is represented. Lobatto implies that the endpoints are included (Eqs. 1 thru 6) (Richard 2019). The GLL quadrature can be stated as:

$$\int_{-1}^{1} f(x)\,dx \approx w_1 f(-1) + w_N f(1) + \sum_i^n w_i f(x_i) \qquad \text{(Eq. 1)}$$

Where:

$$w_i = \frac{2}{N(N-1)[P_{N-1}(x_i)]^2} \qquad \text{(Eq. 2)}$$

$$w_{1,n} = \frac{2}{N(N-1)} \qquad \text{(Eq. 3)}$$

$$P_n(1) = 1 \qquad \text{(Eq. 4)}$$

$$P_n(-1) = (-1)^n \qquad \text{(Eq. 5)}$$

Transformation of the discrete representation of f_n to the space of the Fourier components c_k, e.g. using FFT, is a transformation from physical to spectral space. By increasing the resolution or polynomial order, i.e. number of grid points (N), the error decreases exponentially as opposed to algebraically which is the case for finite-difference methods. "A Fourier series of a smooth function…converges rapidly with increasing N, since the magnitude of the coefficients $|c_k|$ decreases exponentially" (Schlatter 2010). This spectral convergence to the exact solution is a key characteristic of the spectral element method.

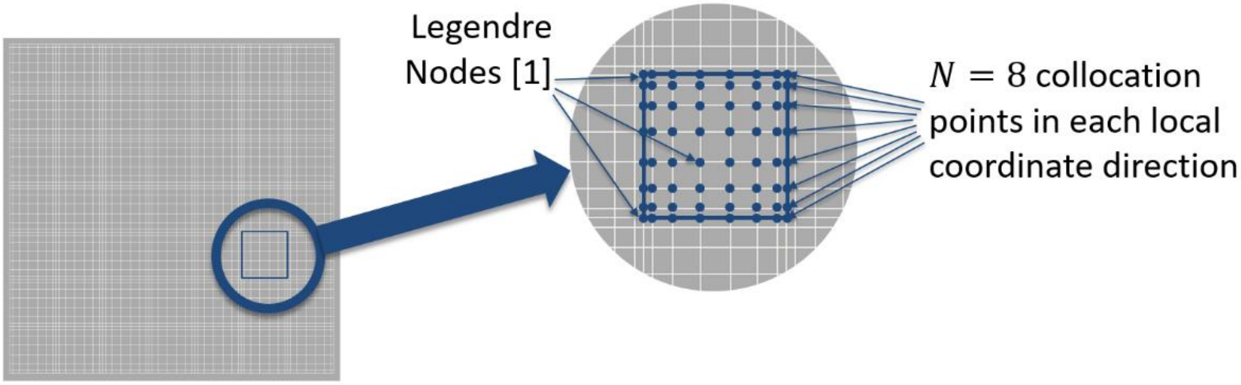

Legendre Nodes [1]
$N = 8$ collocation points in each local coordinate direction

Figure 1: 7x7 Spectral Elements (E=49) with data in terms of 7^{th} order Legendre Polynomial

Temporal discretization in NEK5000 has an implicit and explicit design used for the computation of linear and non-linear terms, respectively. Prior to compilation several temporal variables are defined, e.g. start time, end time and timestep size or CFL. One method is to use a CFL value with a variable timestep, dt. With a variable dt, NEK5000 will use the CFL to determine the upper dt limit. Another method, with greater user influence, is to explicitly define a non-variable dt. A third alternative is to set dt=0 and define the CFL, in which case the program will determine the dt value based on CFL (Argonne National Laboratory 2018).

Meshing in NEK5000 is done using either third-party software or internal utility tools for defining a geometry. This study focuses on using the internal meshing option "genbox". genbox creates a simple single box mesh or a mesh comprised of several boxes. The prescribed boundary conditions are applied to the predefined mesh elements and a new integrated mesh is generated. The use of the following third-party mesh generating software are some of those supported by NEK5000:

- CUBIT + MOAB
- TrueGrid
- Gambit
- ICEM

Computing requirements increase as geometries become more refined, complex and/or polynomial order increases. Scalability is straightforward with the ability to meet simulation requirements on a variety of cluster configurations. Some key factors to improve performance of large-scale simulations include:

- Minimize polynomial order
- Optimize memory allocation
- Maximize timesteps
- Tune user defined settings

Users are additionally able to modify meshes and output files through the use of many subroutines/functions included in the NEK5000 library. Postprocessing quantities are able to be computed by calling subroutines with/and provided functions in the user file. Use of these features requires knowledge of programming in Fortran.

4 Case Studies

The user's first steps are to download, unpack to their home directory and initialize the current NEK5000 release (V17.0). Completing the website's outlined tutorials (Argonne National Laboratory 2018) is extremely beneficial, initiating the user with program flow and file formats needed to develop future cases. The user must make a directory in the location ($HOME/Nek5000/run/) when creating each new simulation/case. A detailed description of an example on how to run NEK5000 can be found in our in-house user manual (Naranjo 2018). Several files from example cases and templates provided in the downloaded tarball are available to help build the new case study's directory, e.g. Case 1's computational domain for two-dimensional (2D) pipe flow is a variation of the geometry file (hillp.box) created during the NEK5000 tutorial. The primary files required to develop new simulations are:

1) *.box – defines simulation geometry, i.e. boundary condition type.
2) *.usr – user control interface via subroutines, i.e. boundary condition values.
3) *.par – defines simulation parameters, i.e. end time.
4) SIZE – defines problem size, i.e. polynomial order.

Details on file structure and various associated information can be found in Appendix 1. Several secondary files are created using the aforementioned files in conjunction with built-in NEK5000 scripts before executing a simulation. The following cases outline the progressing complexity of problems studied in order to develop the AHR natural circulation simulation using NEK5000.

4.1 Case 1: 2-D Pipe Flow with Uniform Heat Flux at the Wall

4.1.1 Problem Definition

Case 1 was created from altering the tutorial's 2D hill problem, the first case a new user develops from templates and by creating new files. The geometry for the computational domain was modified to reflect a length to width aspect ratio of 5. A simple 2D pipe flow with constant heat flux at the wall is the final constructed problem. The initial model, reference Figure 2, was developed without thermal information, i.e. only velocity initial and boundary conditions are applied. The velocity boundary conditions (BCs) at inlet and outlet are defined as Dirichlet velocity (V) and Neumann outflow (O), respectively. Where V is all velocity components of **u** and are explicitly prescribed. O, the no-stress formulation (constant viscosity) of the stress tensor, is defined by:

$$[-p\mathbf{I} + \nu(\nabla\mathbf{u})] \cdot \mathbf{n} = 0 \qquad \textit{(Eq. 6)}$$

Horizontally, the no-slip boundary condition is applied at both walls. Initial conditions match the prescribed boundary condition values where: $u_x = 1\frac{m}{s}$ and $u_y = u_z = 0\frac{m}{s}$. All other aspects of the case files remain unchanged from the tutorial's example, e.g. material properties. Following the initial run, the development of the viscous boundary layer is clearly visible in Figure 2 as velocity increases from zero at the walls to its maximum at center.

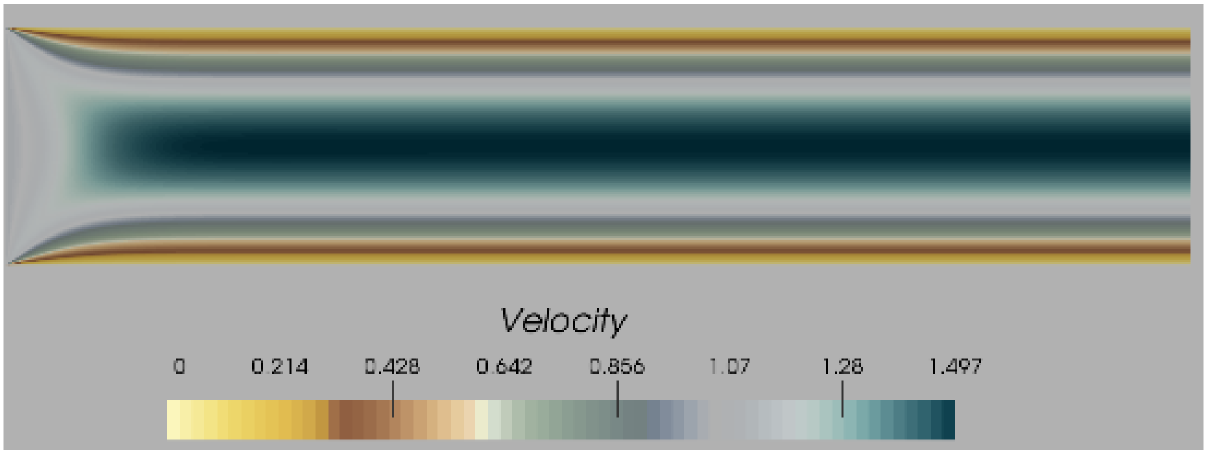


Figure 2: Case 1 Velocity Only

After the successful new simulation was completed, with velocity and isothermal conditions, more parameters were modified and/or added to create a heat flux only scenario. The next step experiments with the case by adding additional and changing existing boundary & initial conditions through the modification of various input files. Thermal BCs at inlet and outlet were specified as Dirichlet temperature (T) and Neumann outflow (O), respectively. Dirichlet temperature conditions require only one specified value, temp, and Neuman outflow for temperature is defined by:

$$k(\nabla T)\cdot\mathbf{n}=0 \qquad \textit{(Eq. 7)}$$

Constant heat flux (f), a Neumann boundary condition, is applied to the walls in the y-direction and is defined by:

$$k(\nabla T)\cdot\mathbf{n}=f \qquad \textit{(Eq. 8)}$$

For velocity, the inlet BC remained Dirichlet but with each velocity component equal to zero. The outlet BC remained Neumann outflow, with no additional parameters or information required. The Fluid in the domain was prescribed to be initially still and at constant temperature, defined initial conditions (ICs) are velocity ($\mathbf{v}=0$) and temperature (temp = 50 °C). The flux boundary (f or q"), applied at the horizontal boundaries is specified as:

$$q"=-2.0\frac{kW}{m^2} \qquad \textit{(Eq. 9)}$$

It is easily seen in Figure 3 the heat flux is negative, i.e. fluid in the pipe is cooled, heat or energy leaves the computational domain.

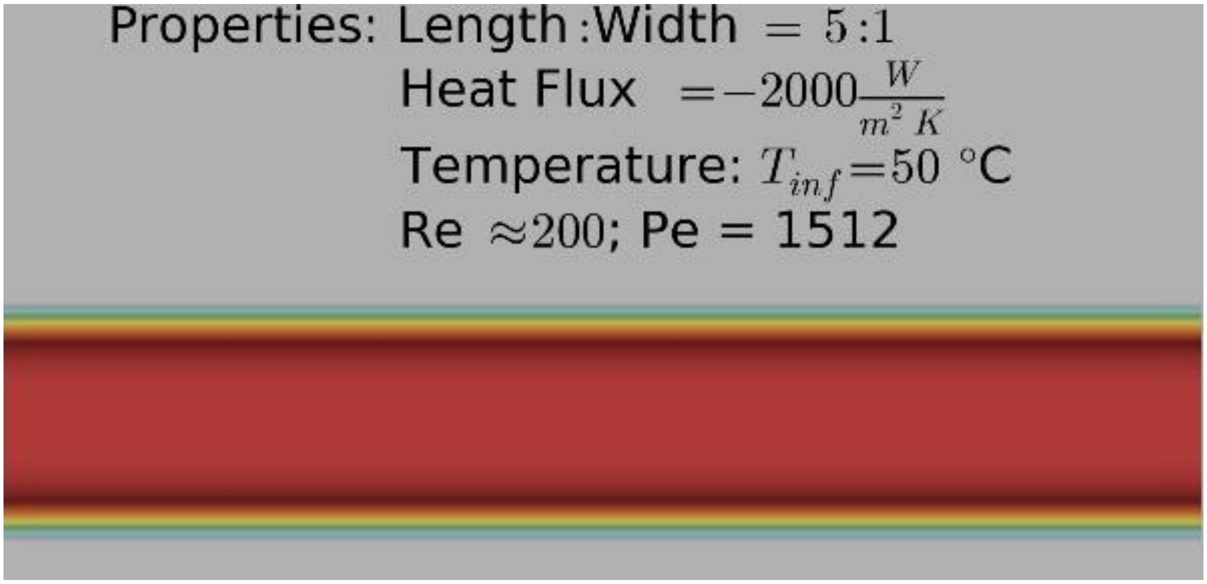


Figure 3: Case 1 Heat Flux Only

After successfully completing the individual runs described above, the velocity & heat flux conditions were coupled and several other modifications were made for the final iteration. The mesh was uniformly refined, the aspect ratio was modified to include both the thermal & hydraulic entrances, and accurate material properties were applied to represent water at 10 ℃ at one bar (MatWeb, LLC. 2018) shown in Table 1.

Table 1: Water Properties at 1 bar (MatWeb, LLC. 2018)

Temperature (℃)	**Density** $(\boldsymbol{\rho},\ kg/\boldsymbol{m^3})$	**Viscosity** $(\boldsymbol{\mu},\ m\boldsymbol{Pa\cdot s})$	**Conductivity** $(\boldsymbol{\kappa},\ W/\boldsymbol{m}\cdot$℃$)$	**Specific heat** $(\boldsymbol{c_p},\ kJ/\boldsymbol{kg}\cdot$℃$)$
10	999.7026	1.307	0.58	4.192

The final version of Case 1 uses the DNS method to solve transient (unsteady-state) two-dimensional water flowing through a Ø1.0 $cm\ x$ 15.0 cm section of heated pipe containing 30 x 210 spectral elements (E=6300), respectively. All data is expressed in terms of a 6^{th} order (N=7) Legendre polynomial, containing EN^2gridpoints = 308,700 or degrees of freedom. There is a positive constant heat flux $(q'' = 1.0\,kW/m^2)$ and no-slip boundary condition applied at the horizontal walls. Velocity ICs and BC's are $\mathbf{v} = u_x =$ 5.0 cm/s at the inlet with an outflow condition at the outlet. Similarly, thermal ICs and BCs at the inlet are temp = 10 ℃, with an outflow condition at the outlet.

Images Figure 4 and Figure 5 show both thermal and hydraulic fully developed flow downstream at 16 seconds. Further detail is shown in Figure 4, focusing on the initial development of the velocity boundary layer at the inlet. Velocity reaches its initial value of 5 $^{cm}/_{s}$ where the corresponding color is transparent, therefore, only the indigo thermal color scheme is visible (highlighted in Figure 4 with black line). Although exact values of boundary layer thicknesses, thermal and velocity profiles are discussed in the next section, it is worth noting the Reynolds number for Case 1 is 382.

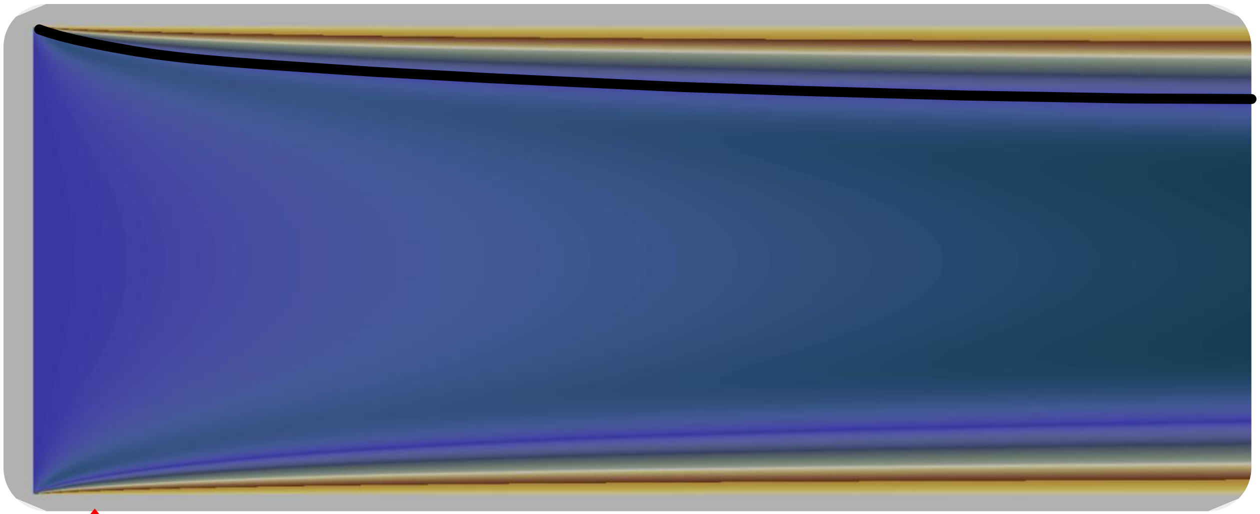


Figure 4: Cutout of Figure 5 with where initial velocity (5cm/s) is transparent and highlighted with blackline

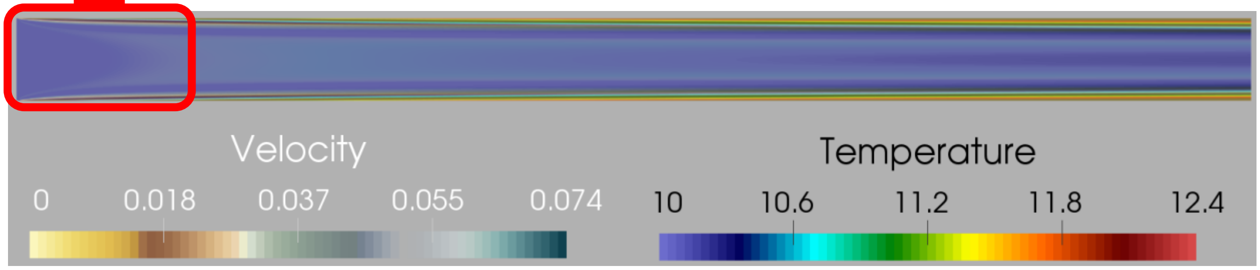


Figure 5:Temperature & Velocity profile of Case 1

4.1.2 Simulation and Analysis

The following analysis of temperature and velocity output is done using cross-sectional data from varying Reynolds numbers in Case 1, at $x = 7.5\ cm$, the axial midpoint. Figure 6 and Figure 7 display both the thermal and velocity profiles, respectively, of the simulation as a function of time. The velocity profile reaches a near steady state condition at the midpoint after 4 seconds, more than three times faster than its thermal counterpart.

As is expected in pipe flow, the thermal profile development is delayed compared to the velocity profile development. This study finds the velocity profile (Figure 7) is fully developed at 4 seconds while the thermal profile (Figure 6) is still developing after 7 seconds. Data for these two figures was extracted from the NEK5000 output files using ParaView.

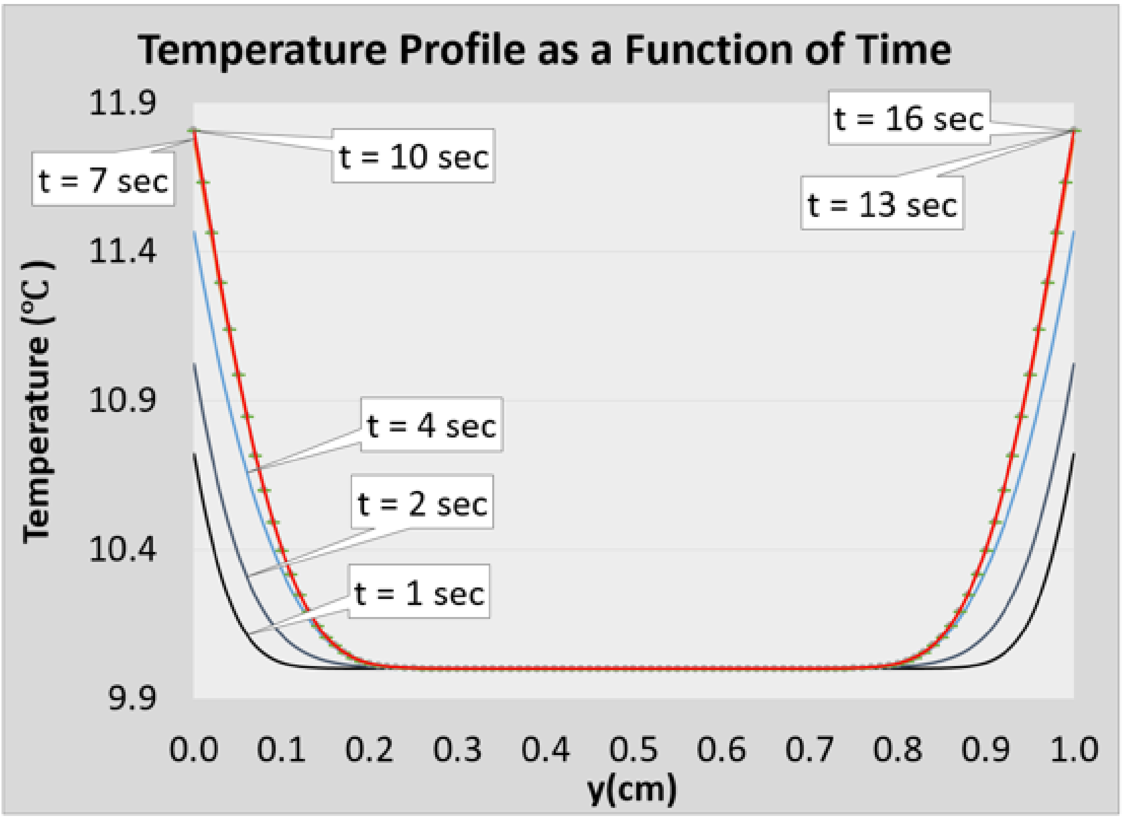


Figure 6: Thermal profile is fully developed at 13 seconds

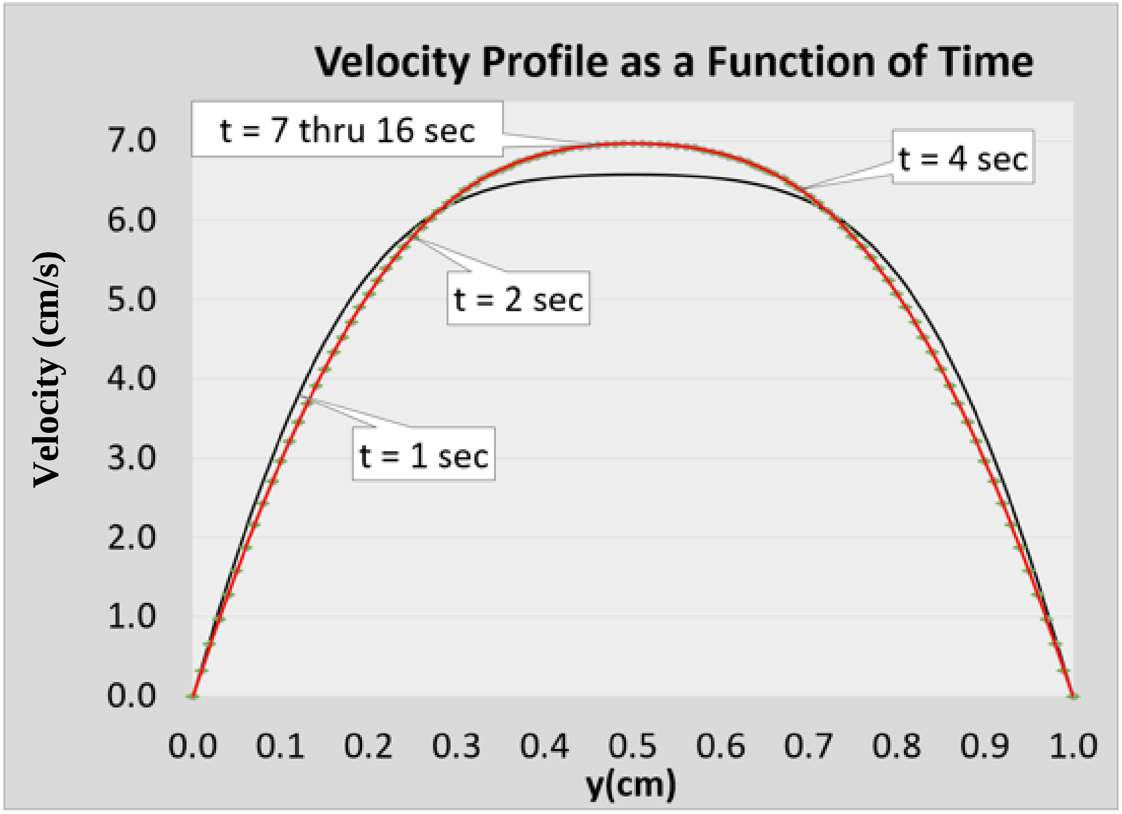


Figure 7: Velocity profile is fully developed at 4 second

While the differences in Figure 6's profiles after 7 seconds are minimal, the data is constant from a time of 16 seconds forward. Herein after, we will report additional analysis at 16 seconds when both velocity and thermal profiles are fully developed.

The methodology used to plot boundary layers requires determining two key thermal hydraulic non-dimensional numbers, the Reynolds and Prandtl number. The Reynolds number (Re_D) is the ratio of inertial to viscous forces, see Eq. 10, where the characteristic length is the diameter of the pipe, D. Inlet velocity is used in our calculations, as the difference between inlet and mean velocity is negligible. From the Re_D we determine the regime in which our pipe flow resides, e.g. laminar, critical or turbulent.

$$Re_D = \frac{inertial\ forces}{viscous\ forces} = \frac{uD}{\nu} \sim 382 \qquad \text{(Eq. 10)}$$

Pipe flow with a Re_D less than 2100 is in viscous or laminar flow. In Case 1 where $Re_D \sim 382 \ll 2100$, therefore the flow is laminar. This leaves determining the Prandtl number (Pr), see Eq. 11, the ratio of momentum to thermal diffusivity:

$$Pr = \frac{momentum\ diffusivity}{thermal\ diffusivity} = \frac{\nu}{\alpha} = \frac{c_p \mu}{k} \sim 9.5 \qquad \text{(Eq. 11)}$$

Using these two non-dimensional parameters, it is possible to calculate the boundary layer thicknesses. Here we report how the Re_D impacts the hydraulic and thermal profile in NEK5000. The datasets used to create Figure 8 and Figure 9 were obtained by varying inlet velocities from 0.05 to 50 m/s corresponding to a range of Reynolds numbers from 382 to 382,400.

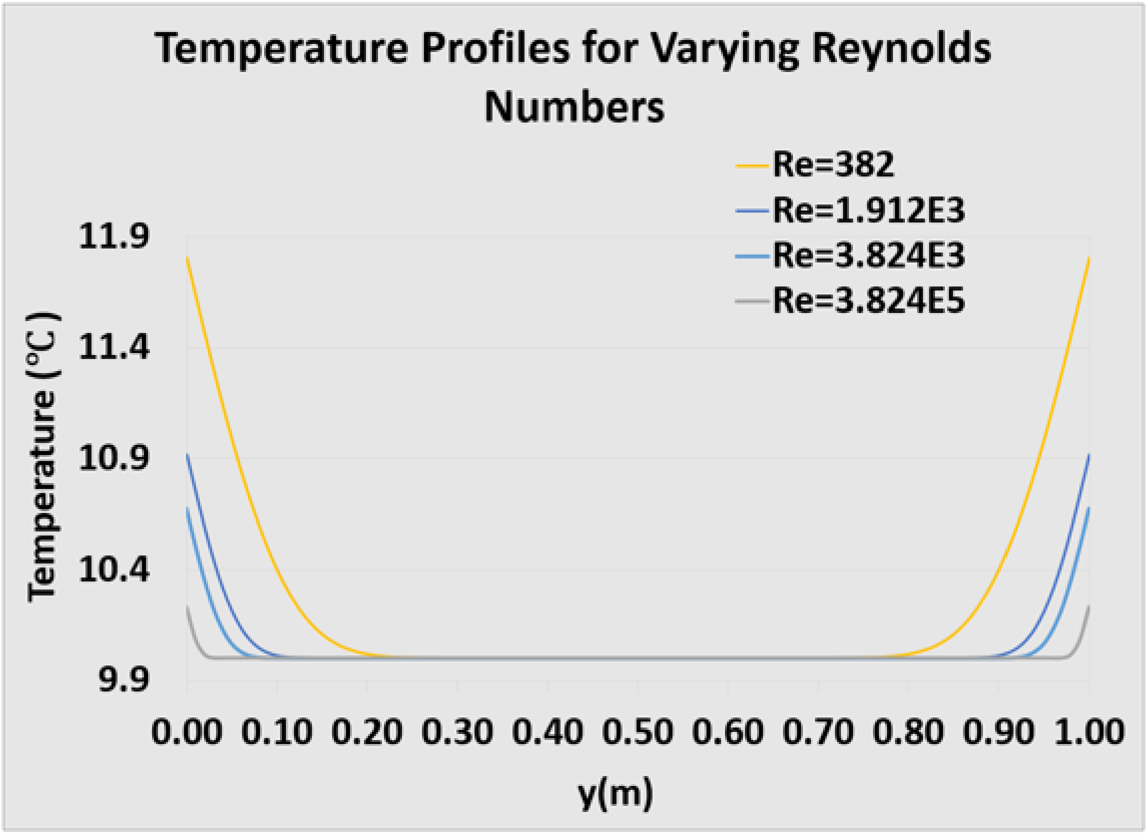


Figure 8: Temperature Profiles

Figure 7 and 8 illustrate how viscous and thermal boundary layers become thinner as the Reynolds numbers increase.

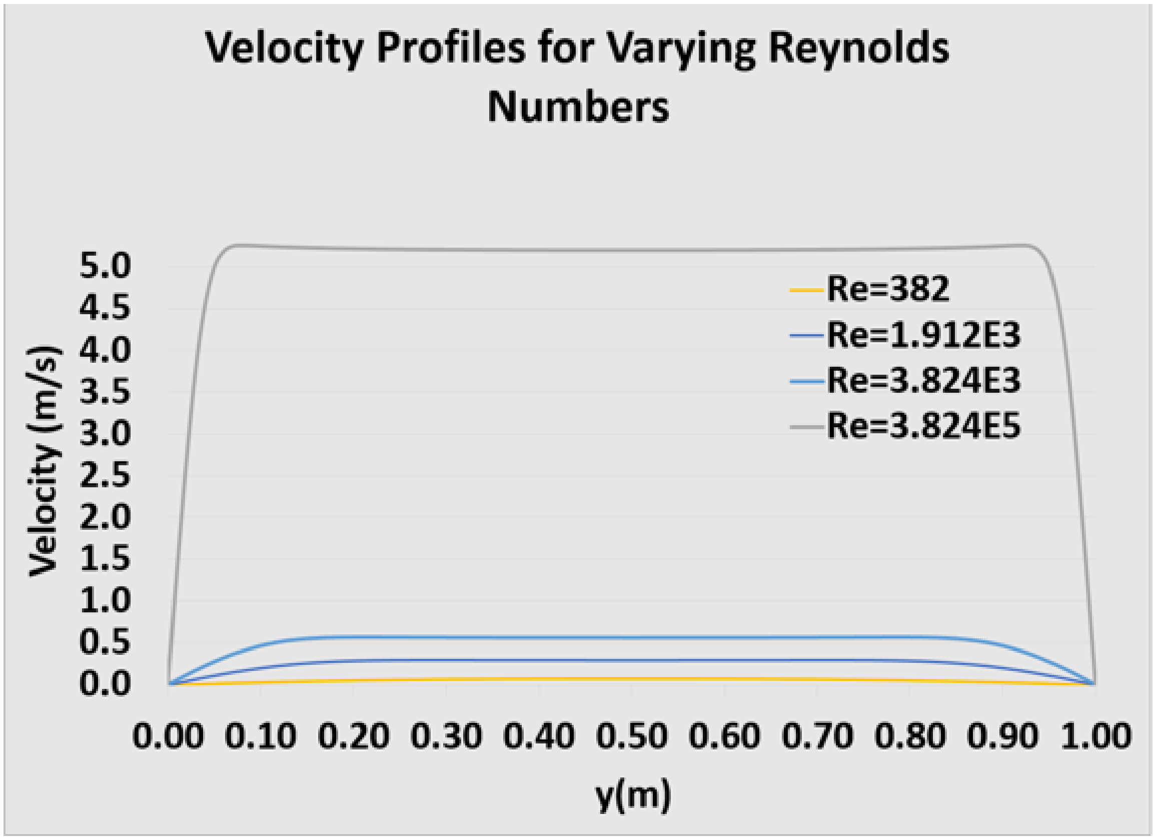


Figure 9: Velocity Profiles

While the above plots are from simulation data alone, Figure 10 illustrates the analytical viscous and thermal boundary layers overlaid on the simulation's cross-sectional data profiles. In Figure 10, we see that the thermal boundary layer is thinner than the hydraulic boundary layer. This is expected when $Pr \gg 1$ and due to the inertial properties dominating the thermal ones. Equations 12 and 13 (Thome and Cataldo 2018) are used to determine viscous and thermal boundary layer thickness, respectively, for laminar incompressible flow along a semi-infinite plate.

$$\delta \approx 4.92D(Re_D)^{-0.5} \sim 0.2517\ cm \qquad \text{(Eq. 12)}$$

$$\delta_{th} \approx \delta Pr^{-1/3} \sim 0.1189\ cm \qquad \text{(Eq. 13)}$$

The boundary layer thickness is generally considered the distance normal to the wall where the velocity or temperature value is approximately 99% of its free stream value. From Figure 10 we see this is true for the thermal boundary layer, since the thermal velocity profile intersects the boundary layer at a value slightly greater than 10 °C. However, in pipe flow the viscous boundary layer thickness continually grows until it reaches the centerline (i.e. the flow is fully developed). When the flow is in the laminar regime, Eq. 12 does not accurately represent the viscous boundary layer, but Eq. 13 does work well for the thermal boundary layer.

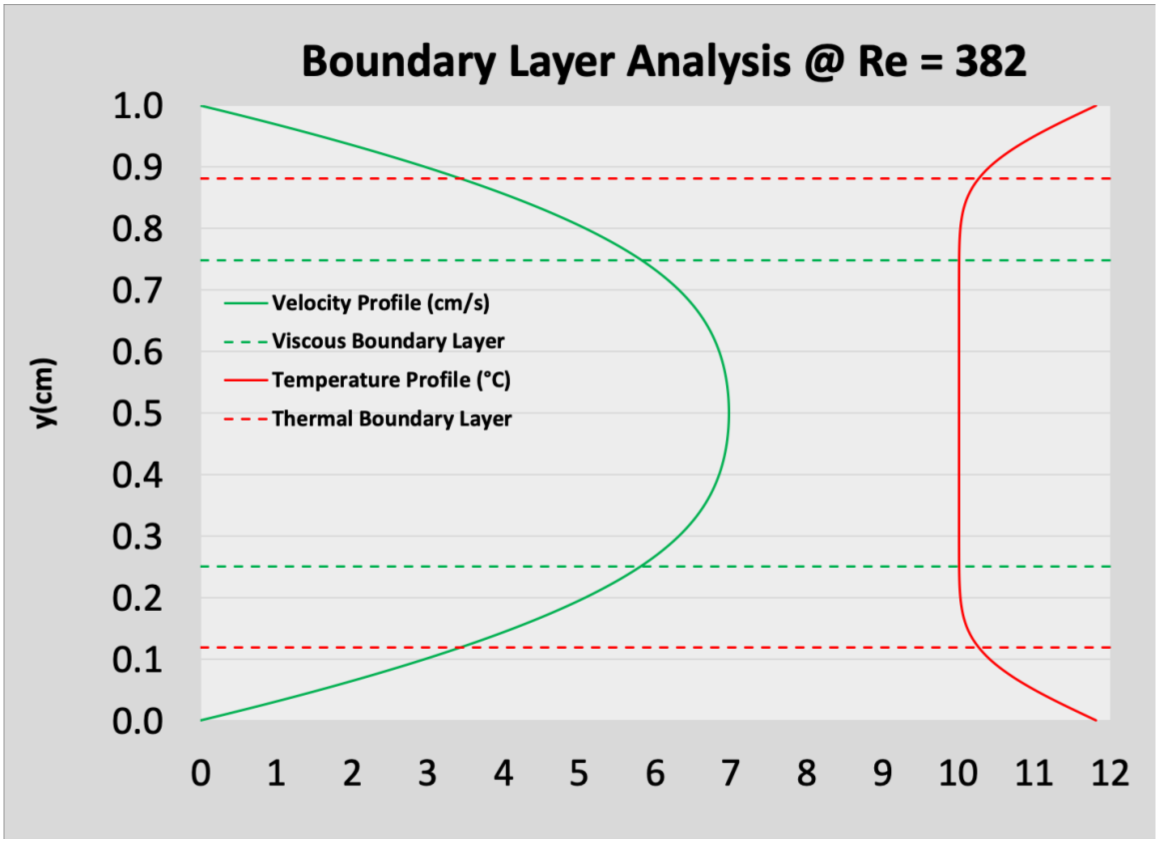


Figure 10: Numerical Simulation Profiles & Analytical Boundary Layers

Having successfully completed an initial simulation and analysis using NEK5000 and ParaView, we next study a case more closely related to our end goal.

4.2 Case 2: 2-D Natural Circulation with Flux Boundary Conditions

Natural convection in a Differentially Heated Cavity (DHC) is a classical heat transfer problem commonly used to compare and evaluate numerical algorithms designed for incompressible recirculating flow (LeQuere 1991). Cases 2 & 3 continue this assessment of NEK5000 by studying the natural convective flow in a DHC of 2 & 3 dimensions, respectively. In our second case study, the physical domain is a $0.25\ m^2$ square with a uniformly distributed mesh in both the x and y-directions. The polynomial resolution within each spectral element is refined to order nine, from order six in Case 1. The simulation contains $7{,}225$ spectral elements of with $\mathrm{N} = 10$, resulting in $EN^2 = 722{,}500$ gridpoints. The heat flux BCs presented below are of three different scenarios: two applied as linear functions of y and one as constant. Resulting in a zero-sum energy balance in each variation, Case 2-1, Case 2-2 and Case 2-3, positive flux is applied to the left wall (i.e. heat source wall) and negative flux (i.e. heat sink wall) is applied to the right. Each variant has an equivalent amount of energy, i.e. heat flux, applied its system.

4.2.1 Problem Definition

All parameters, except boundary conditions, for each of Case 2's subsets are the same. Fluid parameters are the same as Case 1 defined in Table 1 ICs of the domain are velocity ($\mathbf{v} = 0$) and temperature (temp = 10 °C). The Dirichlet no-slip BC is applied at all walls for velocity ($\mathbf{v} = 0$) and Neumann temperature BCs for left and right walls are defined by flux, as follows:

The top and bottom walls are insulated (I):

$$k(\nabla T) \cdot \mathbf{n} = 0 \qquad \text{(Eq. 14)}$$

The left and right BCs are heat flux:

$$k(\nabla T) \cdot \mathbf{n} = q'' \qquad \text{(Eq. 15)}$$

Illustrated below is each of Case 2's BC variants:

4.2.1.1 Case 2-1 (Figure 11):

At the left heat flux is defined as:

$$q'' = (20y - 10)\frac{kW}{m^2} \qquad \text{(Eq. 16)}$$

At the right heat flux is defined as:

$$q'' = 20y\frac{kW}{m^2} \qquad \text{(Eq. 17)}$$

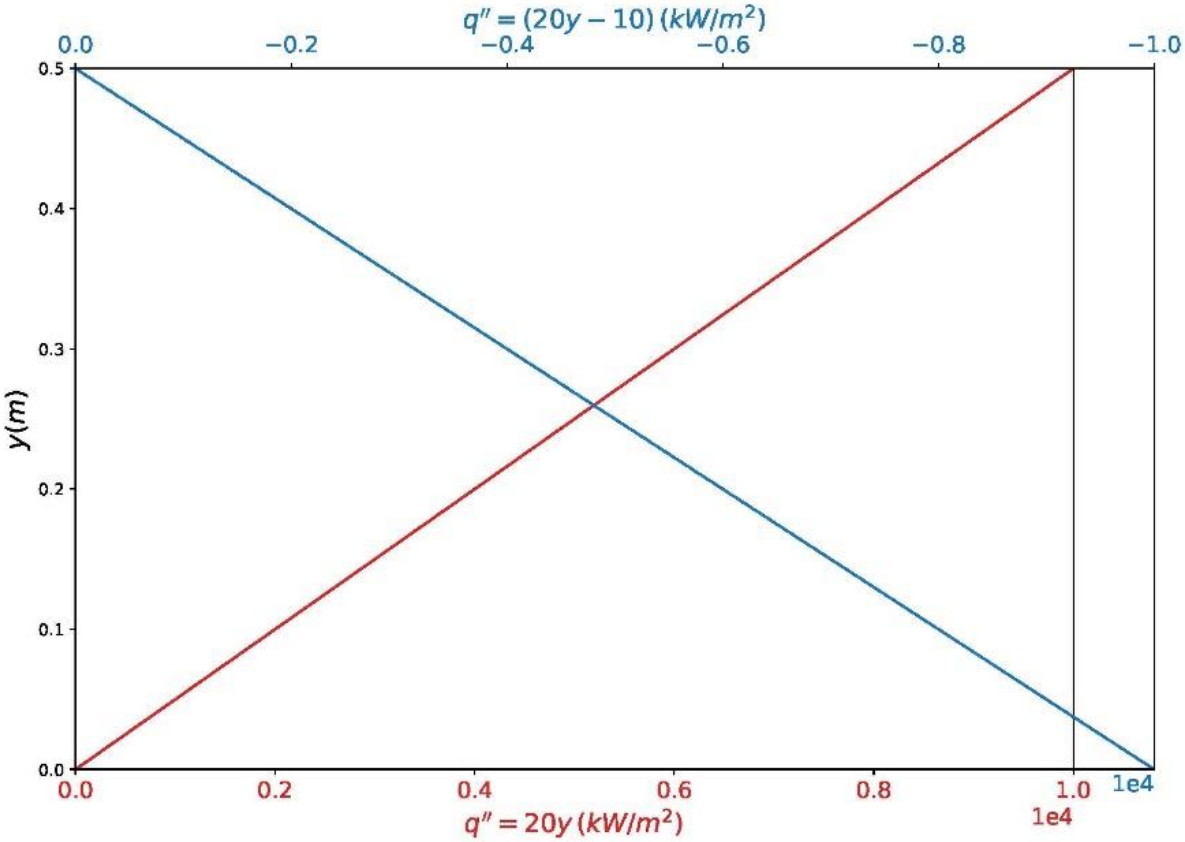


Figure 11: Plot of Case 2-1 Boundary Conditions

4.2.1.2 Case 2-2 (Figure 12):

At the left heat flux is defined as:

$$q'' = 5.0\frac{kW}{m^2} \qquad \text{(Eq. 18)}$$

At the right heat flux is defined as:

$$q'' = -5.0\frac{kW}{m^2} \qquad \text{(Eq. 19)}$$

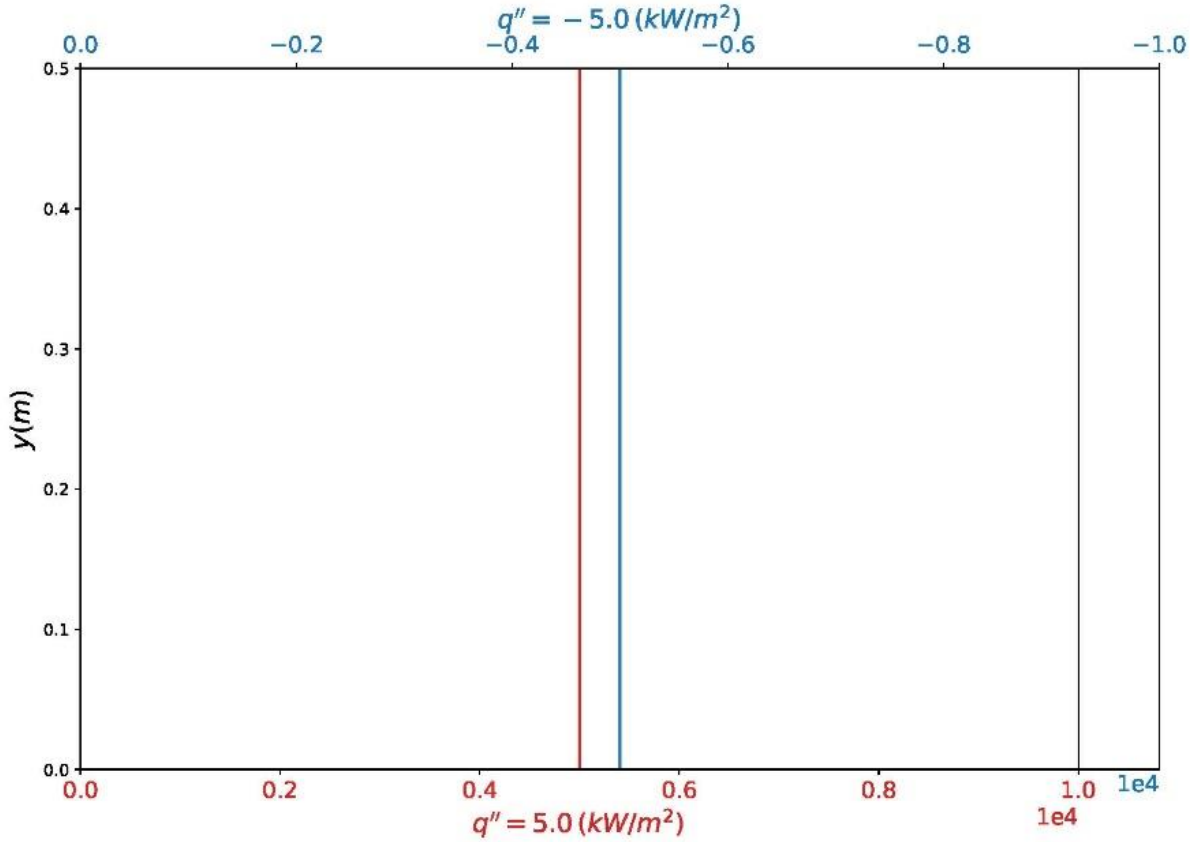


Figure 12: Plot of Case 2-2 Boundary Conditions

4.2.1.3 Case 2-3 (Figure 13):

At the left heat flux is defined as:

$$q'' = (-20y + 10)\frac{kW}{m^2} \qquad \text{(Eq. 20)}$$

At the right heat flux is defined as:

$$q'' = -20y\frac{kW}{m^2} \qquad \text{(Eq. 21)}$$

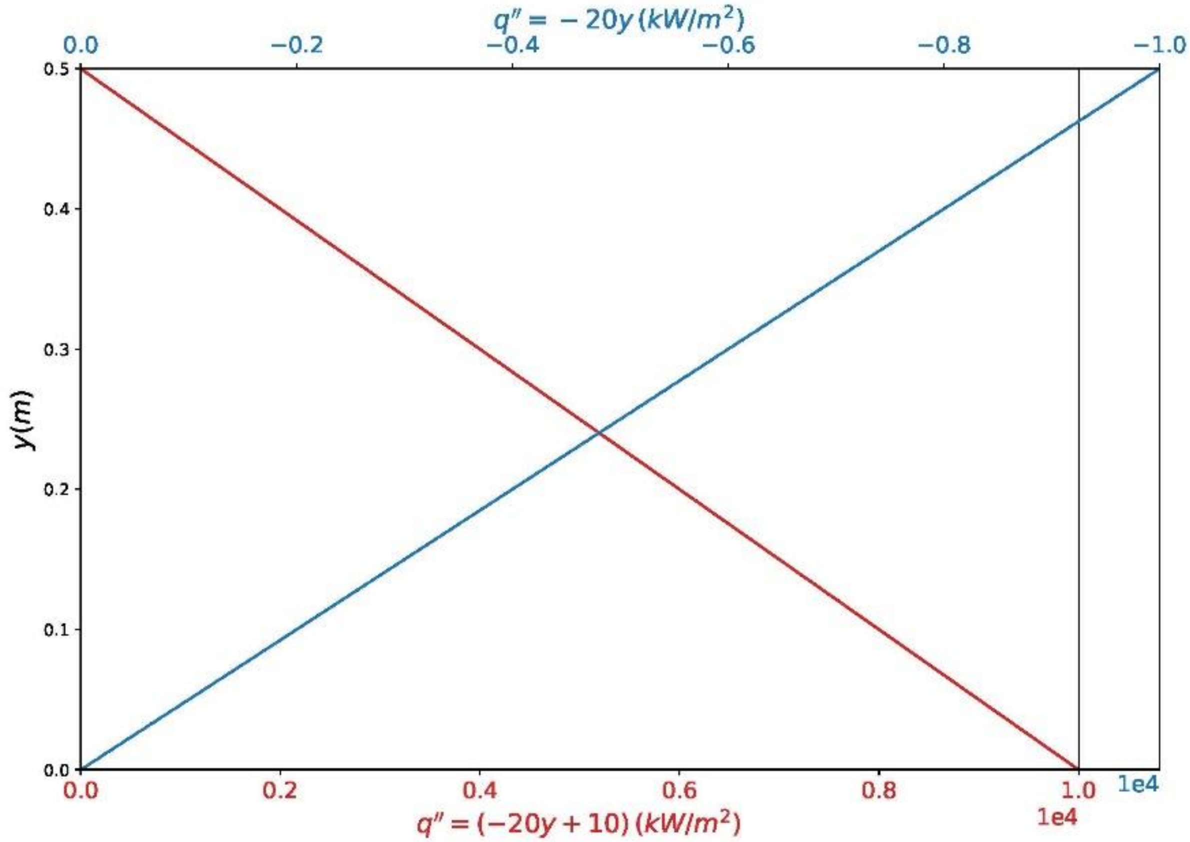


Figure 13: Plot of Case 2-3 Boundary Conditions

For each variant 2.5 kW of thermal power per unit length was introduced to the left wall and extracted from the right wall. Therefore, natural circulation should be triggered if the models are setup properly.

4.2.2 Simulation and Analysis

Different dimensionless numbers categorize natural convective flow. The Grashof number, ratio of buoyancy to viscous forces, is analogous to the Reynolds when studying natural circulation, Eq 22.

$$Gr = \frac{g\beta(T_1 - T_2)x^3}{\nu^2} = 1.26 \times 10^8 \qquad \text{(Eq. 22)}$$

$$GrPr = \mathrm{Ra} = 1.2 \times 10^9 \qquad \text{(Eq. 23)}$$

Generally, the product of the Grashof and Prandtl numbers determine the Rayleigh number, a quantitative means of differentiating the transition from conductive to convective heat transfer. Similar to Eq. 23 a modified form of the Rayleigh number is accepted to characterize the natural circulation flow with flux boundary conditions, see Eq. 24 (Julia C. Mullarney 2004). The transition to convective transport occurs at Rayleigh numbers greater than 1708. Perhaps more importantly, in an enclosed cavity the Rayleigh number additionally characterizes the transition from laminar to turbulent flow at $\mathrm{Ra} > 10^6$ (Markatos and Pericleous 1984).

$$\mathrm{Ra_F} = \frac{g\beta}{\rho_0 c_p \alpha^2 \nu} q'' L^4 = 2.59 \times 10^{12} \qquad \text{(Eq. 24)}$$

Comparison of these two methods of determining Ra, finds a large difference between them. The former falling outside standard empirical correlation boundaries of analysis. After determining the simulation's flow regime via the Rayleigh number, the Nusselt number (Nu) is calculated. The Nusselt number, a nondimensionalized version of the convective heat transfer coefficient, is the ratio of convective to conductive heat transfer. Therefore, convective heat transfer through a fluid will have a Nu greater than 1. Magnitude of the Nu is reflective of the quality of the convective heat transfer, e.g. Nu ranging from 100 - 1000 is considered turbulent flow. The Nu for an enclosed cavity of aspect ratio (A) 1 is calculated via Eq. 25 (Julia C. Mullarney 2004):

$$\mathrm{Nu} \cong Ra^{1/5} = 304 \qquad \text{(Eq. 25)}$$

Therefore, the parameters and conditions applied above result in turbulent convective flow. While natural circulation is driven by changes in mass density, for Case 2 the flow is driven by "thermal density": the body force term in the momentum equation is equal to the varying temperature. Thermal density in this paper is defined by:

$$\mathrm{\rho_T} = \rho * \mathrm{ffx_i} \qquad \text{(Eq. 26)}$$

Where,

$$\mathrm{ffx_i} = \begin{bmatrix} \mathrm{ffx} \\ \mathrm{ffy} \\ \mathrm{ffz} \end{bmatrix} = \begin{bmatrix} 0 \\ temp \\ 0 \end{bmatrix} \qquad \text{(Eq. 27)}$$

The user defined acceleration term components ($\mathrm{ffx_i}$) are multiplied by density (ρ) during the simulation process. This method of setting an acceleration term, ffy, equal to temperature is directly in line with the NEK5000 tutorial example, "Conjugate Heat Transfer" (Argonne National Laboratory 2018).

The figures below allow comparison of the results of each Case 2 scenario near start time:

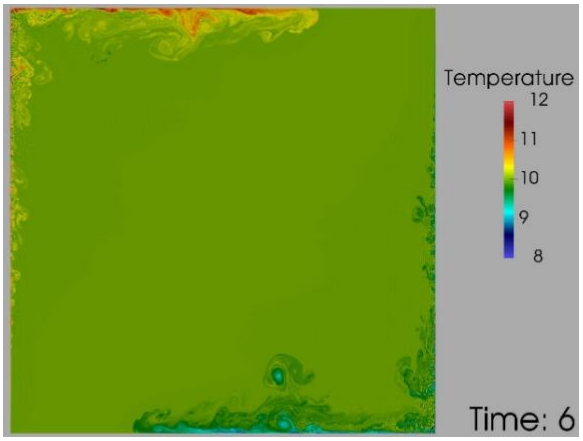


Figure 14: Case 2-1 Simulation Beginning

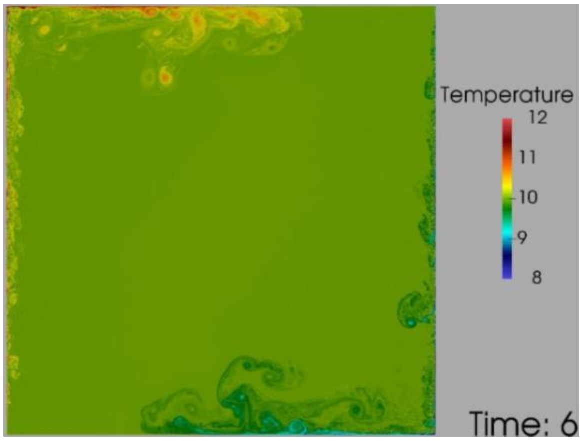


Figure 15: Case 2-2 Simulation Beginning

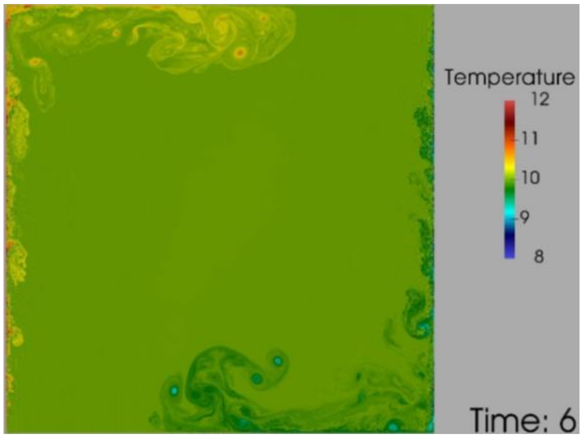


Figure 16: Case 2-3 Simulation Beginning

While slight differences are apparent in the above figures, flow appears increasingly unsteady in Cases 2-2 & 2-3. There is a clearly pronounced impact from the three varying thermal boundary conditions, observed during the latter time steps in the thermal profile development. Specifically, Case 2-1 exhibits nearly stable thermal stratification at approximately one minute (Figure 17). At the same time step, Case 2-2 (Figure 18) begins to exhibit stratified flow and Case 2-3 (Figure 19) remains unsteady throughout the core.

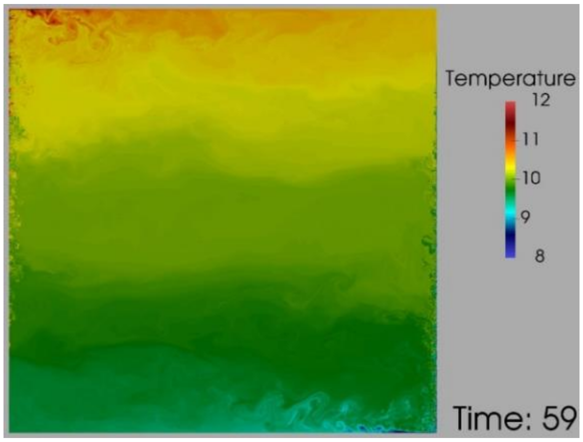


Figure 17: Case 2-1 Thermally Stratified Core

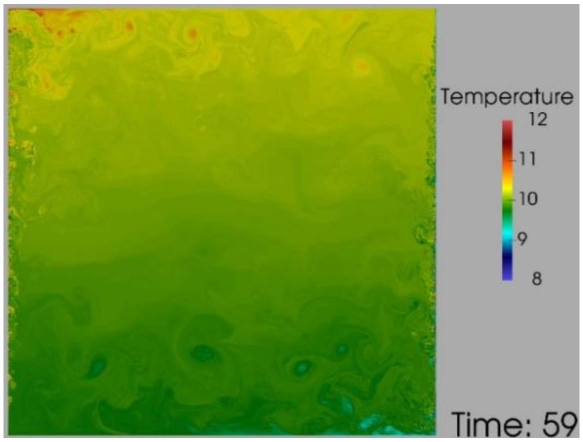


Figure 18: Case 2-2 Some Stratification

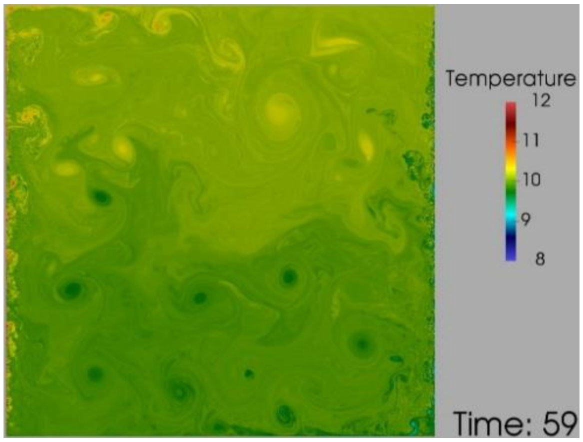


Figure 19:Case 2-3 Mixing Condition at 1 Minute

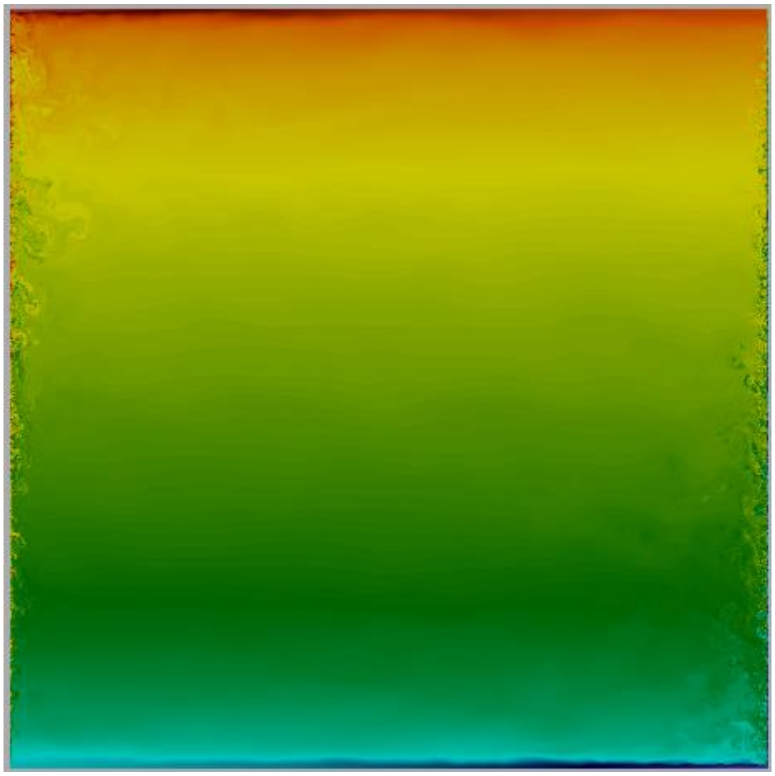


Figure 20: Case 2-2 Thermally Stratified Core at 4 Minutes$(Ra = 1.2 \times 10^9 < Ra_F = 2.59 \times 10^{12})$

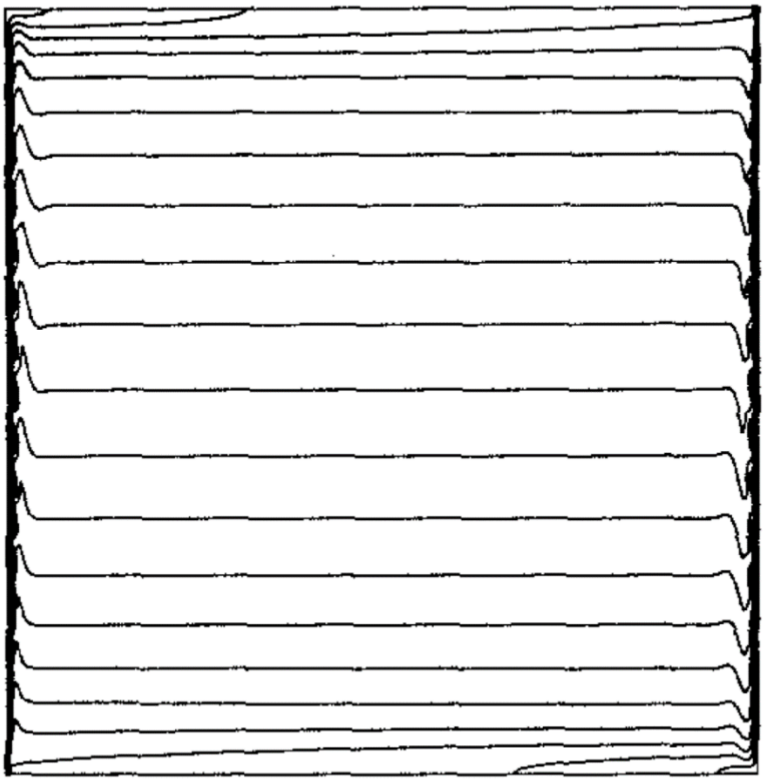


Figure 21: Isotherms for water in the case of adiabatic horizontal walls (Ra = 3 x10^9) (Henkes and Hoogendoorn 1990)

At four minutes time Case 2-2 exhibits linear stability (Figure 20), while Case 2-3 continues to display turbulent mixing (Figure 23). Figure 21 is, qualitatively, in close agreement with Figure 20, with an approximate order of magnitude variation in Nusselt numbers. The method for determining the Ra and Nu in the article Figure 21 is from uses the temperature differential (Henkes and Hoogendoorn 1990). However, from Figure 22 it is clear the geometry of the problem is identical the only variation being the application of energy to the system.

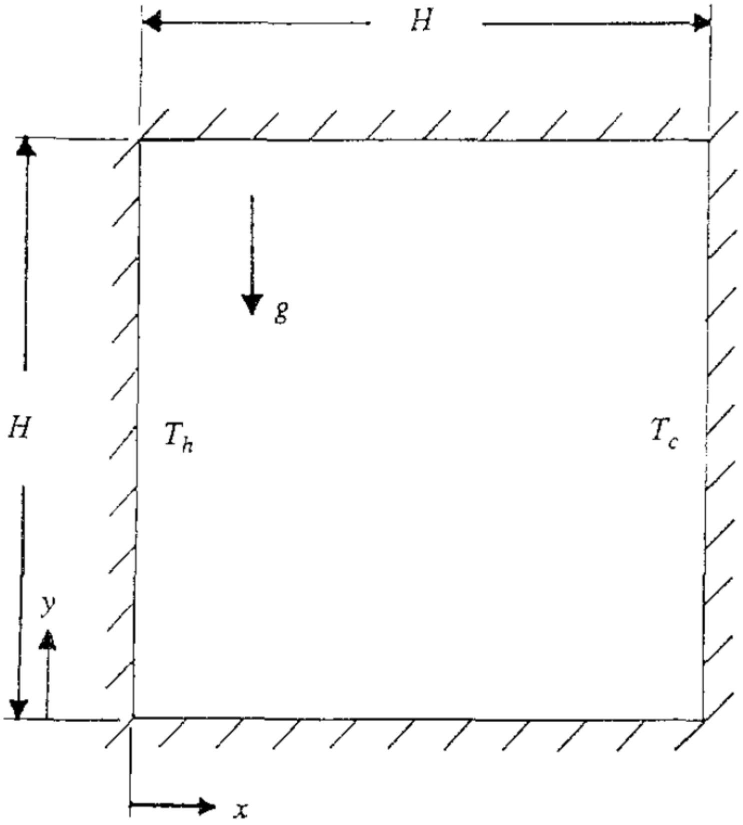


Figure 22:Geometry of the Problem (Henkes and Hoogendoorn 1990)

The initial hypothesis proposes the best mixing conditions will be realized using the boundary conditions prescribed in Case 2-1. In order to test this, Case 2-2 used constant heat flux boundary conditions on an otherwise identical simulation. Following the hypothesis being proven incorrect, it was suggested to invert the boundary conditions of Case 2-1 to accurately represent the natural circulation phenomena as it occurs in most thermal hydraulic systems including heat-source and heat-sink in a closed circulation system, like a nuclear reactor with heat exchangers. After 5 minutes, the longest simulation time, Case2-3's quasi-steady state maintains mixing over the entire domain (Figure 23). Although the denser fluid has settled to the lower half, the solution is unstable with respect to small disturbances and little change has transpired in the four minutes that have passed since Figure 19. This instability against small disturbances is indicative that the Ra is above its critical value, "disturbances no longer decay for increasing time" (Henkes and Hoogendoorn 1990).

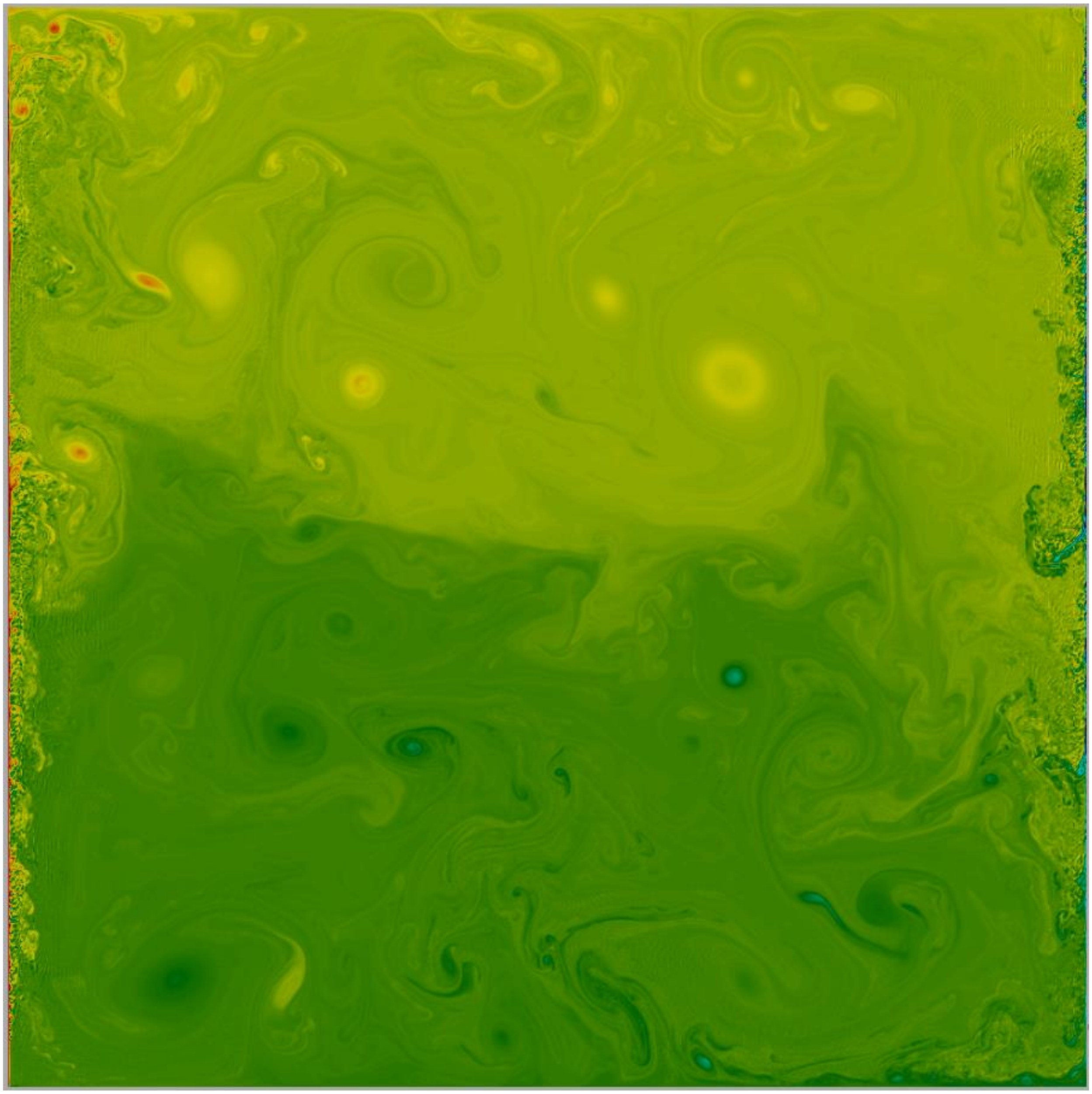


Figure 23: Case 2-3 Mixing at End Time @ 5 min

4.3 Case 3: 3-D Natural Circulation with Flux Boundary Conditions

Case 3 expands upon Case 2-3's simulation by extending the domain into three dimensions, modifying the mesh, geometry dimensions and flux values. Additionally, Case 3-1 makes use of the Boussinesq approximation to drive the flow, as opposed to the thermal density method used in Case 2. A height to length aspect ratio of 1 is maintained, with a depth to height aspect ratio of $0.27\bar{7}$. The result is a rectangular prism domain with a volume of $1.62 \times 10^{-3}\ m^3$. Polynomial resolution within each spectral element remains of order nine $(\mathrm{N} = 10)$. Elemental resolution changed significantly with a finer mesh near the walls and larger elements towards the center. The extension of the domain into another dimension and the mesh refinement, increased the quantity of spectral elements over 500% (E=38,720). The heat flux BCs deviate from Case 2-3 only in magnitude, maintaining a zero-sum energy balance.

4.3.1 Problem Definition

Similar to most of the preceding cases, fluid parameters are defined in Table 1. ICs of the domain are velocity $(\mathbf{v} = 0)$ and temperature (temp = 10 ℃). The Dirichlet velocity no-slip BC is applied at each of the six cavity walls $(\mathbf{v} = 0)$ and Neumann BC, where a heat flux similar to that used in Case 2, is specified. Recalling that the Dirichlet BC specifies the value a solution takes along the boundary and the Neumann BC specifies the value a derivative of the solution takes along the boundary. An example of Dirichlet is defining a value to temperature at the walls, while an example of Neumann is defining a heat flux value at the wall. Below the BCs are outlined:

At the left heat flux is defined as:

$$q'' = (-100y + 18)\frac{kW}{m^2} \qquad \text{(Eq. 28)}$$

At the right heat flux is defined as:

$$q'' = -100y\frac{kW}{m^2} \qquad \text{(Eq. 29)}$$

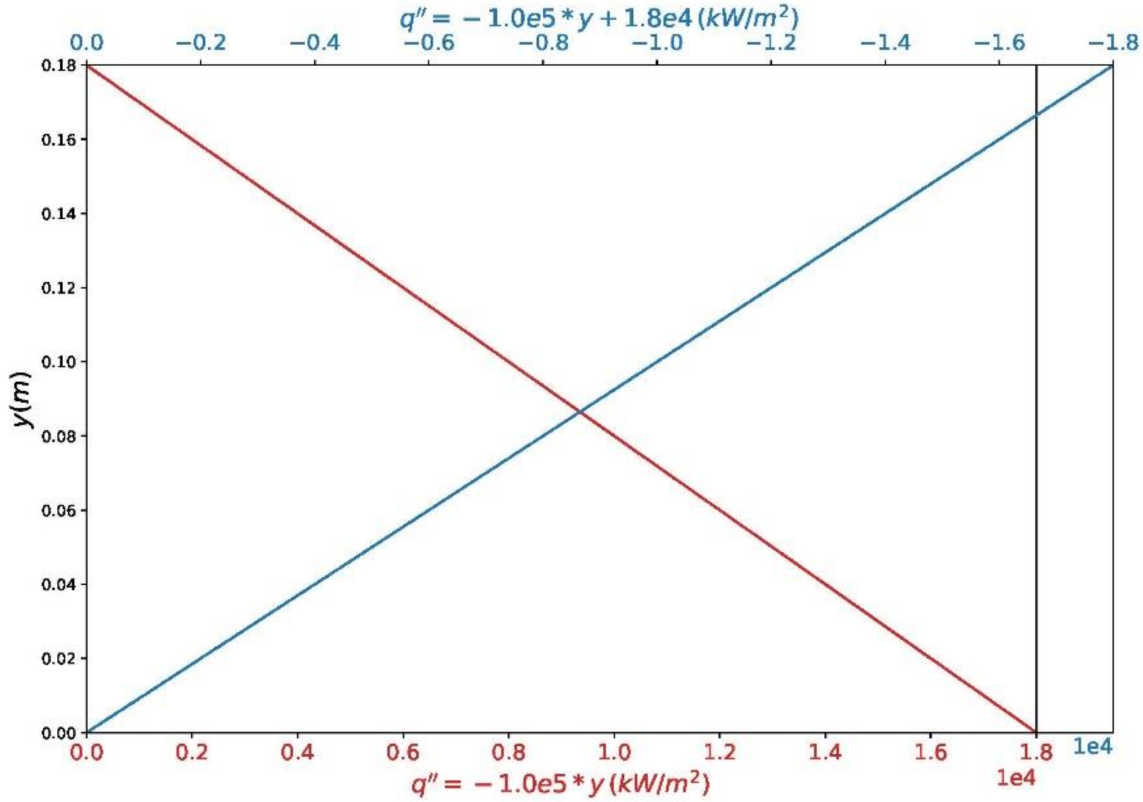


Figure 24: Plot of Boundary Conditions

4.3.1.1 Case 3-1: Boussinesq approximation

The Boussinesq approximation simplifies solving non-isothermal flow by holding density constant everywhere in the Navier-Stokes equations (Eqs 30-32) except where multiplied by the body-force term $(\boldsymbol{f})$. Beginning with the Navier-Stokes Equations for viscous incompressible fluids (Argonne National Laboratory 2018):

Continuity

$$\nabla \cdot \mathbf{v} = 0 \qquad \text{(Eq. 30)}$$

Momentum:

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \mu \, \Delta \mathbf{v} + \rho \boldsymbol{f} \qquad \text{(Eq. 31)}$$

Energy

$$\rho \mathrm{c_p} \frac{DT}{Dt} = \nabla \cdot (k \nabla T) + q_{vol} \qquad \text{(Eq. 32)}$$

Assuming small variations in temperature and density, valid only when $\Delta\rho \ll \rho_0$, the equation of state (Deville 2002) for variable density is defined as:

$$\rho = \rho_0[1 - \beta(T - T_0)] \qquad \text{(Eq. 33)}$$

Where,

$$\rho_0 = \rho(T_0) \qquad \text{(Eq. 34)}$$

The body-force term is equal to the acceleration due to gravity $\boldsymbol{g}$ for natural convection.

$$\rho \boldsymbol{f} = \rho \boldsymbol{g} \cong \rho_0 g_i[1 - \beta(T - T_0)] \qquad \text{(Eq. 35)}$$

Resulting in the following ‘Boussinesq equations’:

$$\nabla \cdot \mathbf{v} = 0 \qquad \text{(Eq. 36)}$$

$$\rho_0 \frac{D\mathbf{v}}{Dt} = -\nabla p + \mu \, \Delta \mathbf{v} + \rho_0 \boldsymbol{g}[1 - \beta(T - T_0)] \qquad \text{(Eq. 37)}$$

$$\rho_0 \mathrm{c_p} \frac{DT}{Dt} = \nabla \cdot (k \nabla T) + q_{vol} \qquad \text{(Eq. 38)}$$

The body force term as defined above is equivalent to ffx in the user file’s subroutine “userf”, and will herein after be referred to as the body force term. “The buoyancy forces that give rise to the free-convection currents are called body forces” (Holman 2010). The userf subroutine multiplies each body force term, ffx$_i$, by density. Therefore, defining ffy = $g[1 - \beta(T - T_0)]$ applies the Boussinesq approximation to Case 3-1 using the correct gravitational orientation ($g \approx -9.81\, m/s$) and volumetric thermal expansion coefficient ($\beta = 87.97 \times 10^{-6}\ ℃^{-1}$).

4.3.1.2 Case 3-2: Variable “Thermal Density”

The body force term in Case 3-2 is identical to that of Case 2, i.e. ffy = temp. Including this Case 3 variation in this study allows for the investigation of the effects of different body force conditions on the simulations.

4.3.2 Simulation and Analysis

The investigation into the differences between Case 3 variants primarily depends on qualitative analysis and begins with finding similarities. Figure 25 and Figure 26 illustrate the initial turbulent flow of Case 3-1 and Case 3-2, respectively, along the vertical walls. The Boussinesq case provides a clearer rendering of the fluid’s thermal distribution. It is believed this is caused by a better heat transfer through convection, as a function of time, in case 3-1. While both share the same material properties, the buoyancy force’s impact, as a function of thermal density, on the circulation in Case 3-2 does not allow for adequate thermal convection to be realized in a visual rendering. The thermal density approach of Case 3-2 does impose a more rapid increase in buoyancy force than Case 3-1. Boussinesq flow (Case3-1) is an established extremely accurate approximation of buoyancy-driven flow for many cases found in nature. The eddies formed as the flow near the wall becomes turbulent is similar in both figures, although flow development occurs 12 times faster in Case 3-2. With temperature multiplied by density, non-Boussinesq case, the rate at which the buoyancy driven flow develops is much greater than with its Boussinesq counterpart.

Referring to Eqs. 24 & 25 the Rayleigh and Nusselt numbers, respectively, for Case 3 are:

$$\mathrm{Ra_F} = \frac{g\beta}{\rho_0 c_p \alpha^2 \nu} q'' L^4 = 7.82 \times 10^{10}$$

$$\mathrm{Nu} \cong Ra^{1/5} = 151$$

Where the initial parameter value, ρ_0, is prescribed at the fluid's initial temperature in each case. This establishes the fluid is at rest across the entire computational domain at t = 0 and a consistent reference point for each similar case study.

While to a lesser degree than Case 2 (Nu $\cong 304$), the above calculation shows the flow in Case 3 is also turbulent.

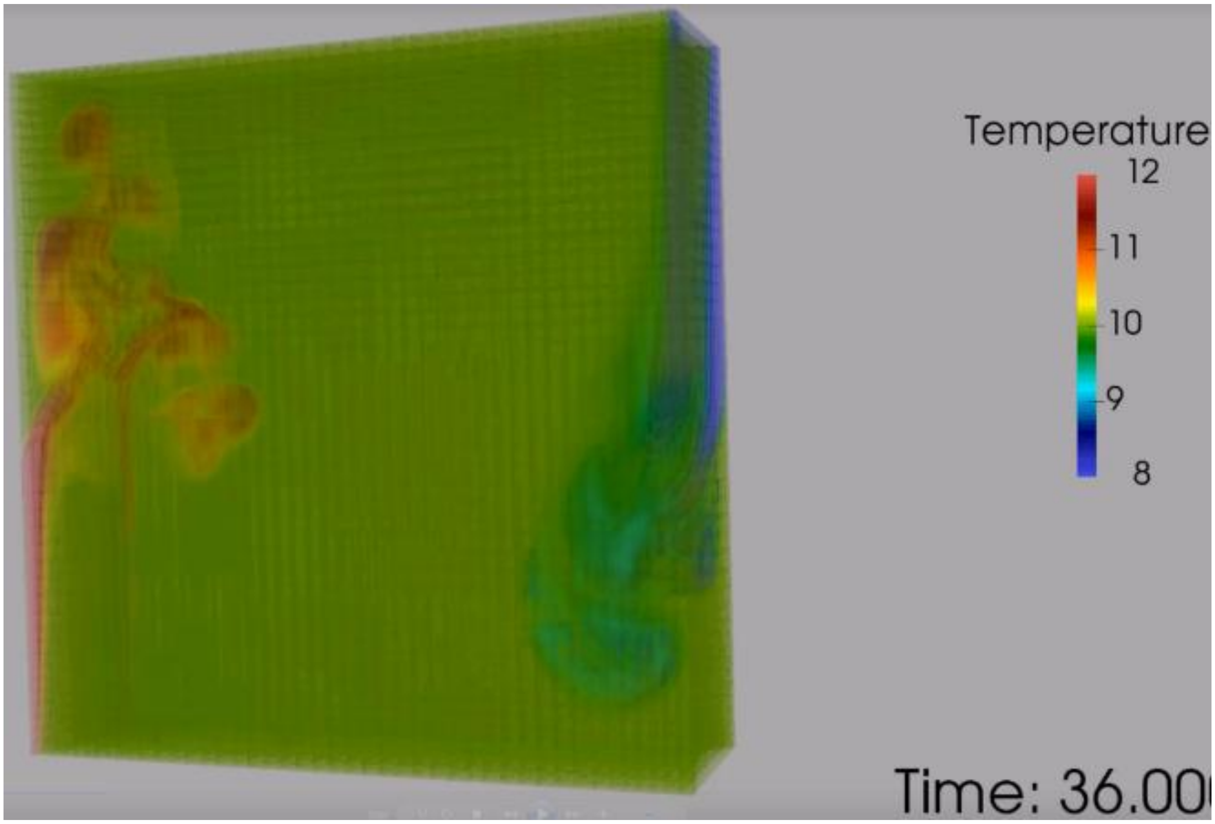


Figure 25: Case 3-1 Vertical Turbulence at 36 seconds

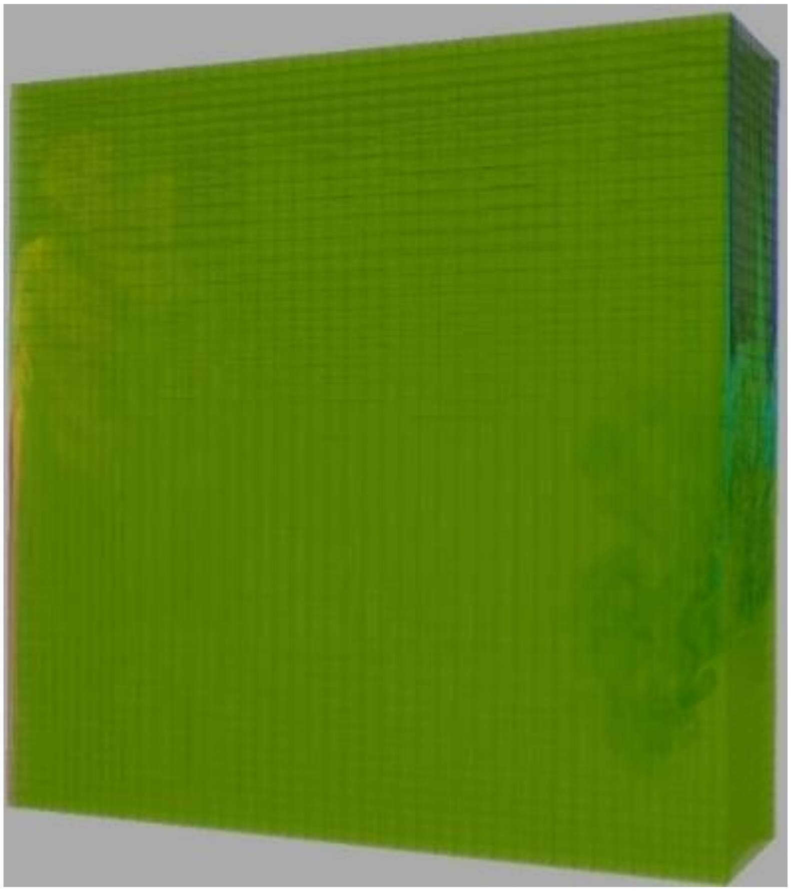


Figure 26: Case 3-2 Vertical Turbulence at 3 Seconds

Below in Figure 27 and Figure 28, the time proportionality constant of 12 produces a similar qualitative flow development in each case. Corresponding with the two figures above, the thermal density method results in a smaller temperature variation.

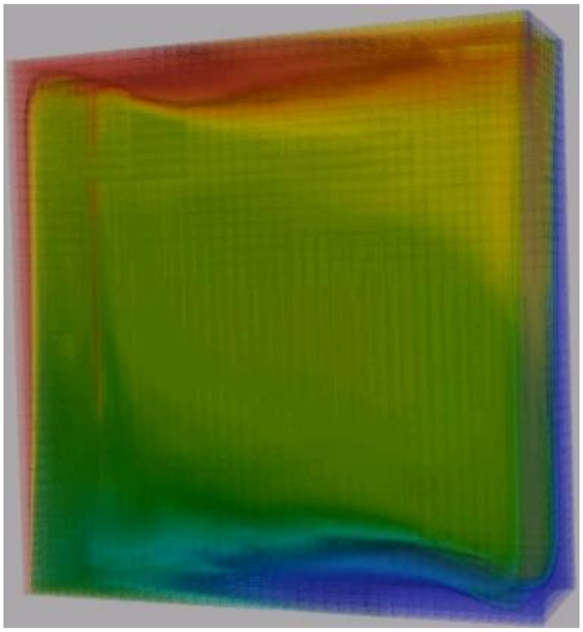


Figure 27: Case3-1 @ 1 minute 48 seconds (12 x 9 sec)

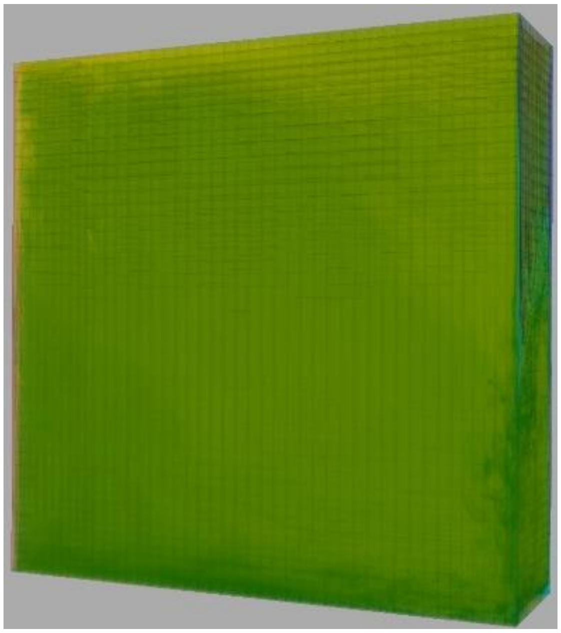


Figure 28: Case3-2 @ 9 seconds

The difference in computational time needed to arrive at each case's results is significant. Case 3-1 (Boussinesq approximation) ran to its prescribed end time of 180 seconds in approximately 1.5 days. Conversely Case 3-2 (thermal density) terminated at a simulation time of 12.5 seconds after 10 days of computational time, the cluster's maximum wall time. Although the thermal density method was very useful in initiating the author with NEK5000, the Boussinesq approximation will be used henceforth, were applicable. This is due to its greater thermal convection, reputation for accurate approximation of natural flow and computational stability.

5 Case 4: Natural Convection in an AHR

Based on the experience gained developing previous case studies with NEK5000, a natural circulation model for a representative AHR is configured. Driving natural circulation in this model is convective Boussinesq flow and an energy balance between heat generation & heat sink. Case 4's simulation focuses on the prediction of a quasi-steady-state thermal equilibrium condition in the numerical model. In addition, the current model captures the natural circulation behavior in a single-phase domain which deviates from a radiolysis induced two-phase natural circulation phenomenon in an actual AHR system. It is believed the two-phase modeling capability in NEK5000 is still under development, so the work scope for Case 4 is narrowed down to single phase only. Fluid properties for the fuel solution are presented in Table 2. The computational domain is a rectangular prism with a circular extrusion radially centered and geometric aspect ratios as follows:

- Length to Width $= 1.0$
- Length to Height $= 0.2$
- Length to Diameter $= 2.5$

Resulting in an overall volume of $3.5 \times 10^{-2}\ m^3$ where Figure 29, Figure 30 and Figure 31 display the mesh and geometry. Case 4's mesh has a more refined resolution near the walls, with larger elements centered between the cylindrical channel and outer walls (E=96,000). One of the most challenging aspects of Case 4 is its mesh construction. The meshing methodology applied for Case 4 is very different from that used during the development of previous cases. Each previous mesh was rectilinear, where the mesh for the current case uses a cylindrical/Cartesian transition. Additionally, this mesh development involves using the "legacy" method of compiling the simulation, *.rea file. Prior to the initialization of the element's boundary locations, the shape of the elements is defined as either box, octagon, or circle illustrated in Figure 29. The are several methods available for geometry definition in the NEK5000, here we use the simplest one with three different element types used to define the control volume mesh. Although the mesh and geometry are initially created using cylindrical coordinates, interestingly it is converted into Cartesian coordinates prior to solving. In addition to this radical change in the domain creation process compared to the simpler design in previous case studies, there is also a marked change in thermal conditions. Heat generation changed from being applied at the walls to a volumetric heat generation rate where a constant temperature is applied at the vertical walls as cooling surfaces. The heat generation rate or power density of the solution fuel is prescribed as $0.05\frac{kW}{L}$ in this case study.

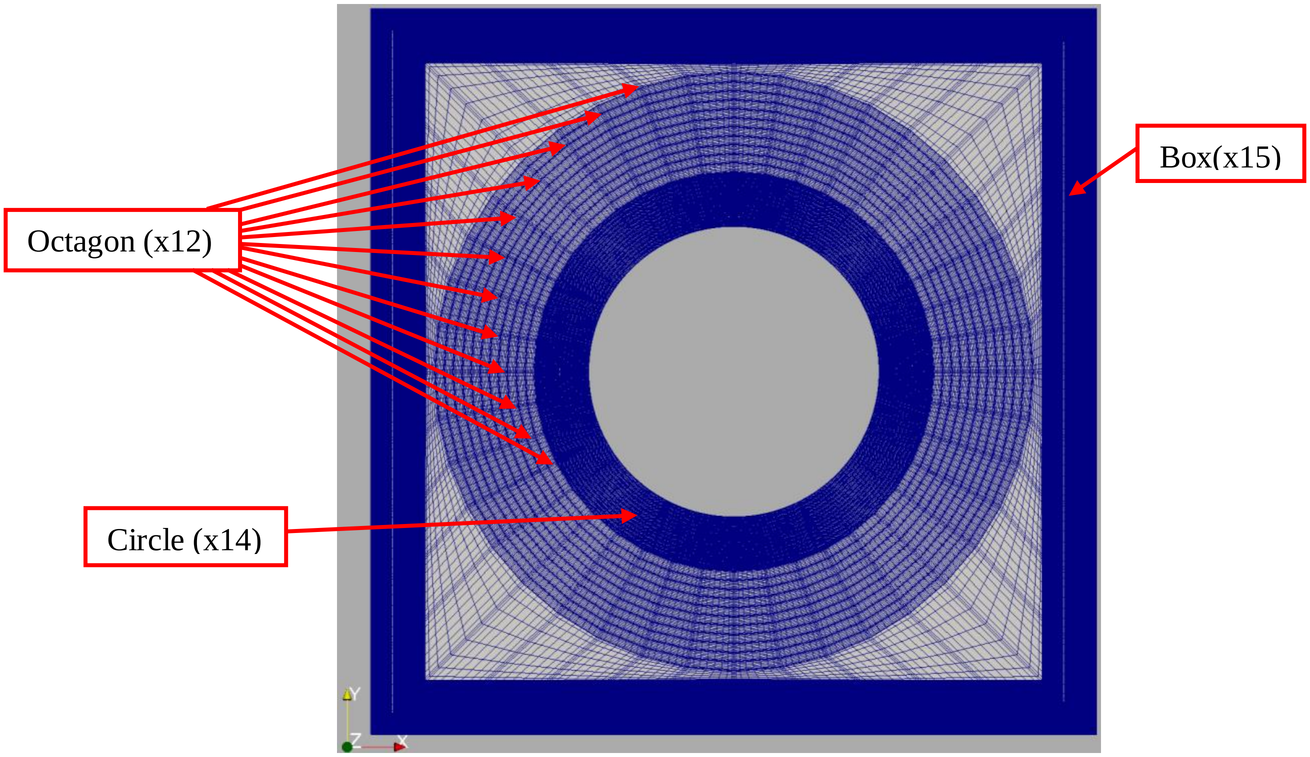


Figure 29: Mesh in the xy-plane

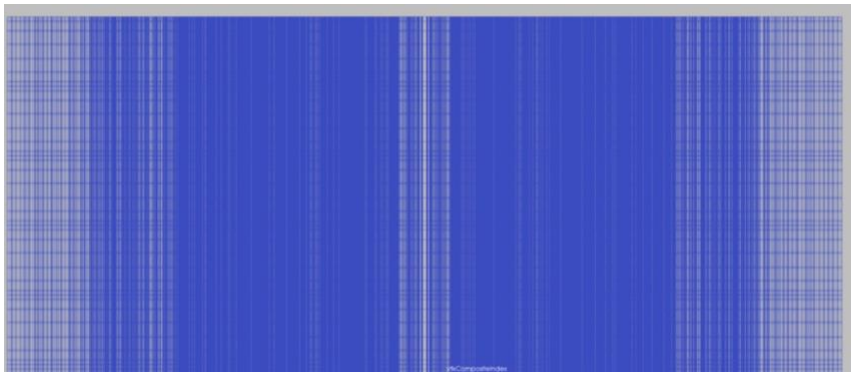


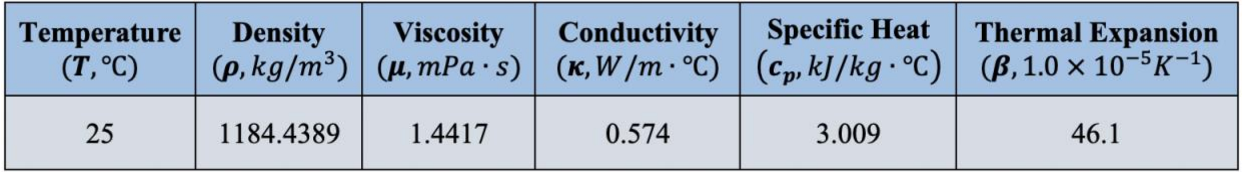
Figure 30: 2-D Enlarged View of xz-plane

Figure 31: 2-D View of xz-plane

5.1 Problem Definition

In each Case 4 variant, all parameters are unchanged with the exception of polynomial order. The solution's fluid parameters were calculated based on a correlation developed in a previous milestone report (Kim and Buechler 2016) and are outlined in Table 2 at the initial temperature. ICs of the domain are velocity ($\mathbf{v} = 0$) and temperature (temp $= 25$℃). No-slip ($\mathbf{v} = 0$) BC is applied at each of the rectangular prism's six walls, in addition to the inner circular wall. A constant temperature BC (temp $= 25$℃), is applied at the inner circular wall and the four outside walls extending the height of the domain. At the xy planar extents of the domain (i.e. top and bottom surface), the Neumann insulated (I) BC is applied ($q" = 0$). The power density or volumetric heat generation rate, q_{vol} , from the energy equation (Eq. 33) is initialized at $50\ kW/m^3$. A description of how and where these conditions are applied is summarized in Appendix 1 below.

Table 2: Uranyl Sulfate Properties (Kim and Buechler 2016)



Temperature (T, ℃)	Density ($\rho, kg/m^3$)	Viscosity ($\mu, mPa \cdot s$)	Conductivity ($\kappa, W/m \cdot$ ℃)	Specific Heat ($c_p, kJ/kg \cdot$ ℃)	Thermal Expansion ($\beta, 1.0 \times 10^{-5} K^{-1}$)
25	1184.4389	1.4417	0.574	3.009	46.1

Polynomial resolution within each spectral element is the differentiating factor in each Case 4 subset: $N = [6, 8, 10]$ corresponds to a Legendre Polynomial order of $[5, 7, 9]$, respectively. Increasing polynomial order refines the spectral resolution within each element, thereby increasing computational requirements.

5.2 Simulation and Analysis

Defining Case 4's geometry and boundary conditions were the most difficult obstacles to overcome during this study (reference Appendix 1.1). The domain is outlined using cylindrical coordinates, explicitly defining elemental geometry. This involves prescribing where in the domain the center is located, elemental shapes and domain extents. Radially the values began with a cylinder wall and end with a "box" wall. Azimuthally the domain is uniformally divided between 0 to 2π. Exponential convergence on the exact solution is accomplished by increasing polynomial order (N-1). The effects of changing this parameter extend beyond the accuracy of the solution to:

5.2.1 Output File Size

Increasing the mesh's polynomial resolution (reference Appendix 1.2 Figure 49, line 13, 'lx1') results in a significantly larger output file. Table 3 illustrates the impact polynomial order has on the output file size. Note that there was no significant output file size difference between DNS solver and LES solver. Therefore, by minimizing the polynomial order, storage space is optimized.

Table 3: DNS & LES Polynomial Order's Impact on File Size

N	6	8	10
File Size (GB)	0.64	1.5	2.9

5.2.2 Computation Time

Required simulation runtime increases as polynomial resolution increased while using 30 nodes, 12 processors per node. Computational time (runtime) required to reach 127 seconds (simulation time), illustrated in Table 4, presents the impact polynomial order has on computational runtime for each Case 4 variant. Further analysis shows simulations with a polynomial of order 5 have approximately 1.6x faster runtimes than simulations of order 7, while setting $N = 6$ and 8 show approximately 4.9x and 3.0x faster runtimes than when N=10, respectively. To obtain these relational values, consider the following example:

$$\frac{N_{10}}{N_6} = 4.8571 \Rightarrow 1.351 * 4.8571 = 6.5620$$

Table 4: DNS-Polynomial Order's Effect on Runtime

N	6	8	10
Computational Runtime (Days)	1.351	2.173	6.562

5.2.3 Precision of Solution

Reducing N by one-step (literature recommends using even numbers only) decreases both the output file size and computational runtime by approximately 50% on average. Until now, the information correlating accuracy to polynomial order has been from literature review only (Argonne National Laboratory 2018). When N=6 the simulation terminates after 127 seconds (Figure 33), although the prescribed end time is 180 seconds. After reviewing the simulation of N=8 immediately following 127 seconds, a hypothesis was formed that N=6 is too coarse for DNS calculation, therefore causing the simulation to terminate. Below are renderings of the simulation's progression with the computational domain bisected at the center of the geometry (0.1, 0.1) normal to the xy-plane.

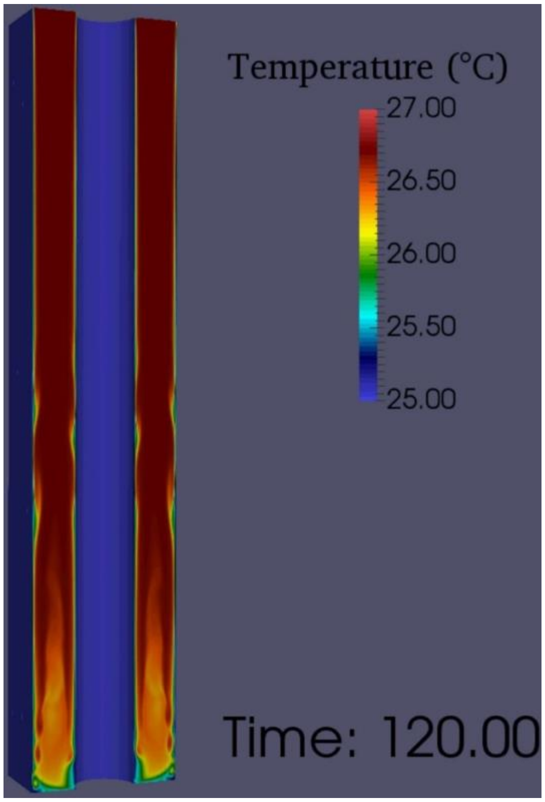


Figure 32: Rayleigh-Taylor Instabilities along Walls

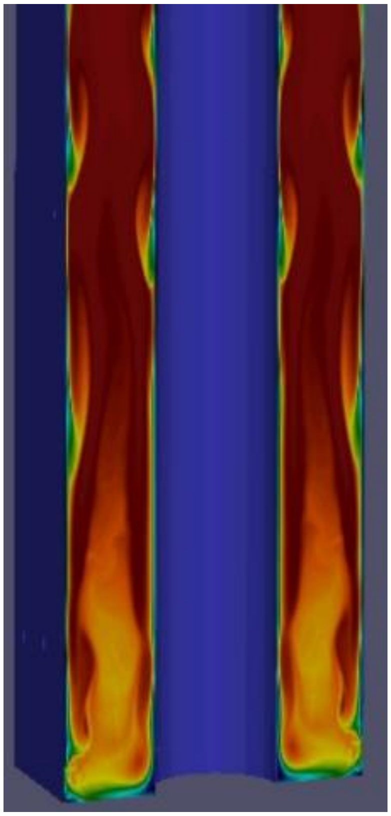


Figure 33: Termination Point of N=6 @ t = 127

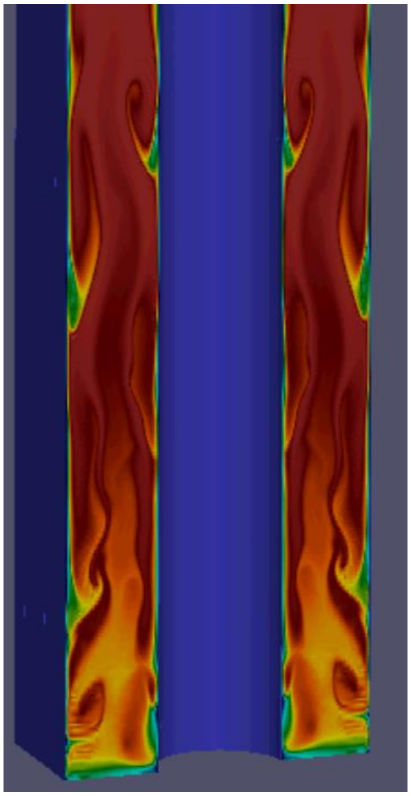


Figure 34: Rayleigh-Taylor&Kelvin-Helmholtz Instabilities @ t = 134

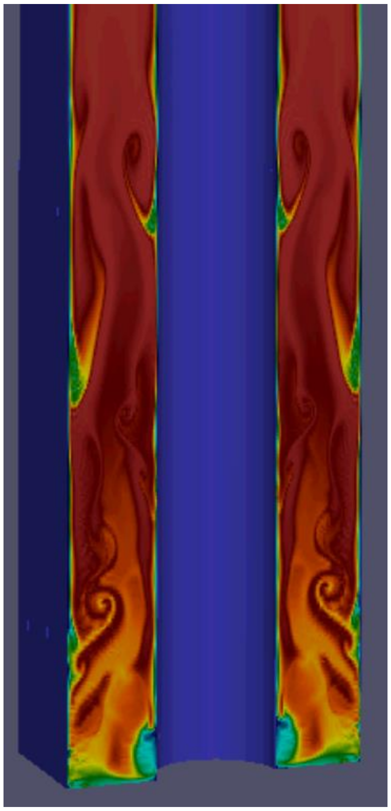


Figure 35: Fully Developed Kelvin-Helmholtz Instabilities @ t = 136

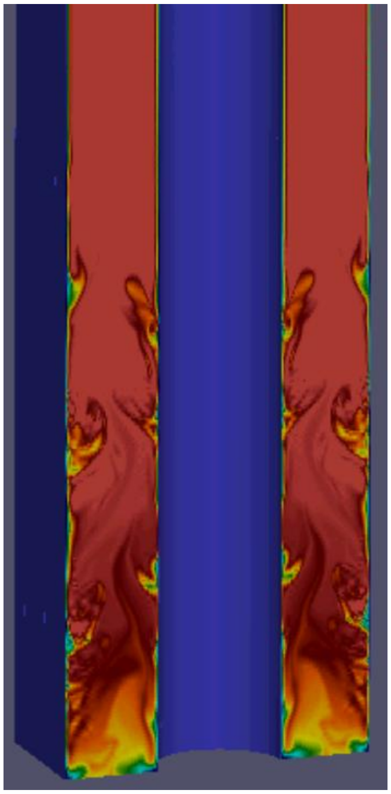


Figure 36: Prescribed End Time @ t = 180

The Rayleigh-Taylor instability (RTI) originates at the interface of fluids with different densities. The interface is unstable if the lighter fluid pushes the heavy fluid. Conversely, the interface is stable if the heavy fluid pushes the light (Sharp 1984). Kelvin-Helmholtz instabilities (KHI) develop along the interface of the fluids at later calculation stages when the fluids density ratio is sufficiently small and the Reynolds numbers is large enough (Daly 1967). An additional criterion of the Kelvin-Helmholtz instability is “the interface between two fluids is unstable if there is a jump in the tangential component of the velocity across the interface" (Sharp 1984). It is widely accepted that the RTI is a variable density instability, and KHI is shear velocity induced turbulence instability. Interestingly, these well-defined instabilities are observed in our high-resolution transient calculations. The relationship between the two instabilities is best summarized as “*In general, it is not possible to separate the two principles for fluid motions because they interact. However, Taylor instability is more important in the case of radically different densities, whereas Helmholtz instability is more important for interfaces separating fluids of nearly equal density…*” (Birkhoff 1954). While our flow is Boussinesq, i.e. density is constant and gravitational acceleration fluctuates, we will assume buoyancy is driven by fluctuations in density herein after. Figure 32 and Figure 33 show the development of sinusoidal-like RTIs along the walls at different time steps. In this gravitational field, natural Boussinesq flow, the RTI forms at the interface of the sinking denser solution near the cooling walls and the rising less dense heating fluid region between the walls. KHIs develop and become visible after approximately 130 seconds simulation time, as upward and downward fluid velocities create sufficient destabilizing shear effects which overcome the stabilizing effects of stratification (Vujinovic and Rakovec 2015), reference Figure 34, Figure 35 and Figure 36.

Below in Figure 37 and Figure 38, using velocity magnitude values, are renderings corresponding to the above time steps where examples of both hydraulic instabilities are clearly depicted. Figure 37 shows the velocity profile's stable and unstable RTI fluid interface along the vertical boundaries. Shown in Figure 38 is the fully developed KHI velocity profile. Sharp concludes his report stating the belief that "a Taylor unstable interface is also subject to Kelvin-Helmholtz instability". In Figure 39 velocity in the z-direction is plotted across a line bisecting x and approximately two-fifths up from the base (z=0.395). Referencing Figure 40, we see the developing sinusoidal-like RTIs growing from the top to the bottom, becoming unstable, and KHIs are shown in the lower half of the temperature and velocity figures corresponding to 136 seconds simulation time. While quantitatively unverified at this time of this report, the author believes our current study with NEK5000 illustrates a qualitative transition from RTI (early stage density induced turbulent mixing) to KHI (later stage shear induced mixing).

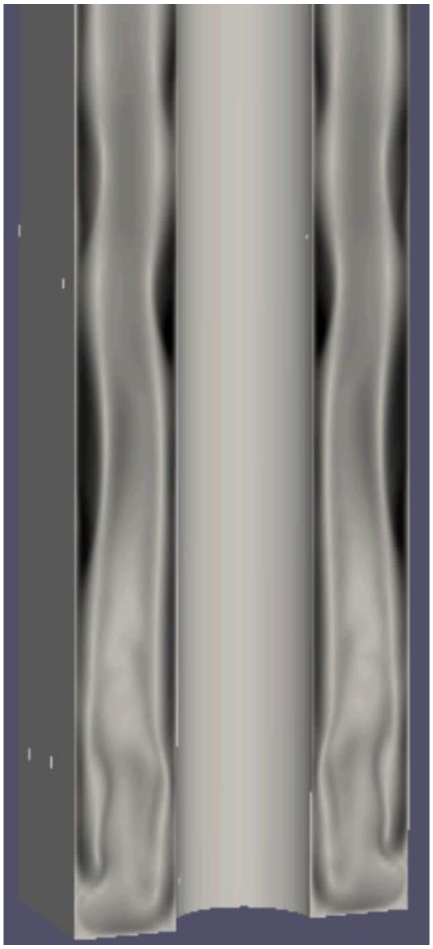


Figure 37: RTI evolution @ t = 127 seconds

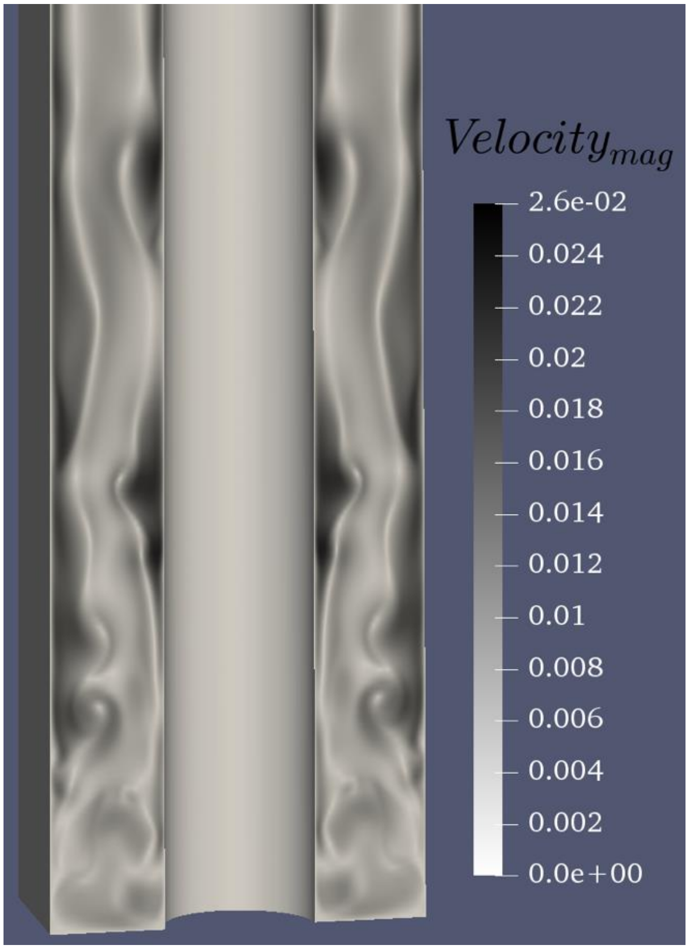


Figure 38: KHI Develop @ t = 136 seconds

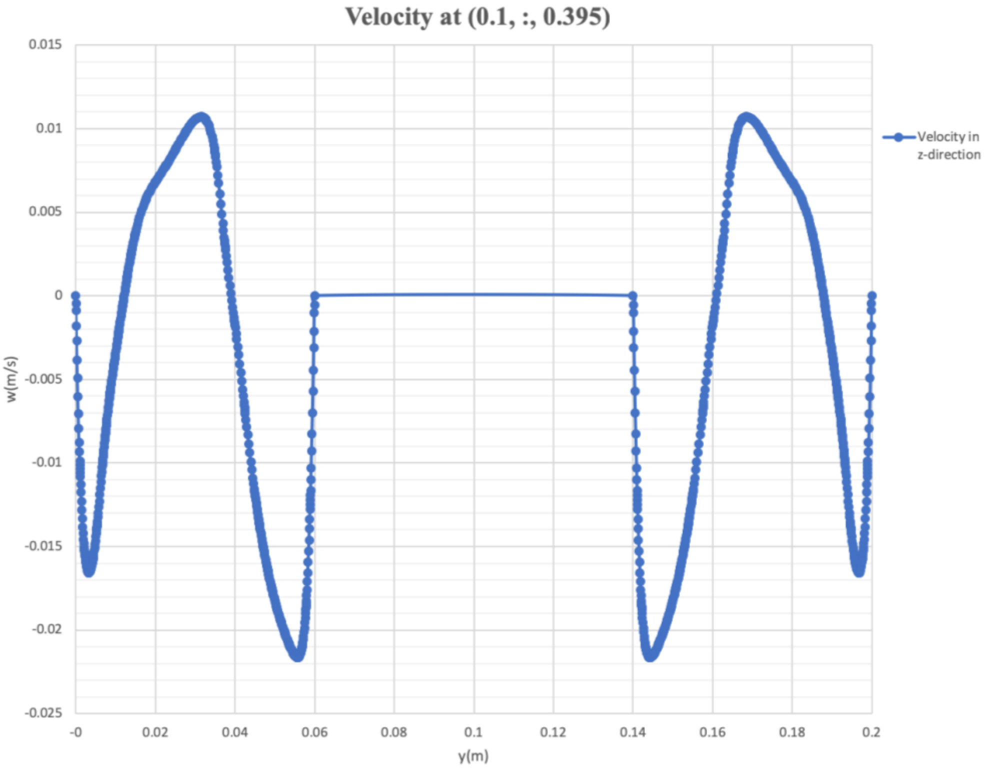


Figure 39: Velocity in the z-direction at t = 136 seconds

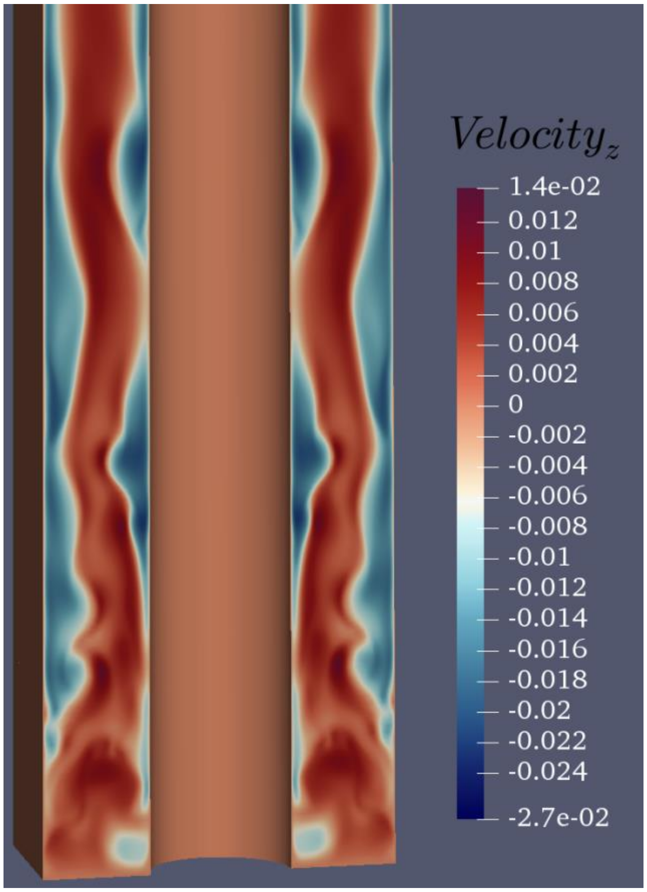


Figure 40: Vertical Velocity Components at t=136 seconds

Experimental results find the N=8 simulation is the only case to run successfully through to the prescribed end time (t =180 seconds). With insufficient resolution being the hypothesis for the N=6 simulation's divergence resulting in an incomplete run, the cause for N=10 simulation's incompletion was exhausting the maximum allotted single run wall time of NEN-5's local cluster (Pete). To address the issue of large computational requirements, moving forward simulations make use of NEK5000's restart feature. Analysis of the volume-averaged solution temperatures extracted from the three different spectral resolution cases tested (i.e. N6, N8, N10), indicate polynomial order has minimal effect on solution precision, depicted in Figure 41 and Figure 42. The caveat being, the thermal analysis is restricted to the timeframe preceding divergence of the coarser polynomial resolution's solution.

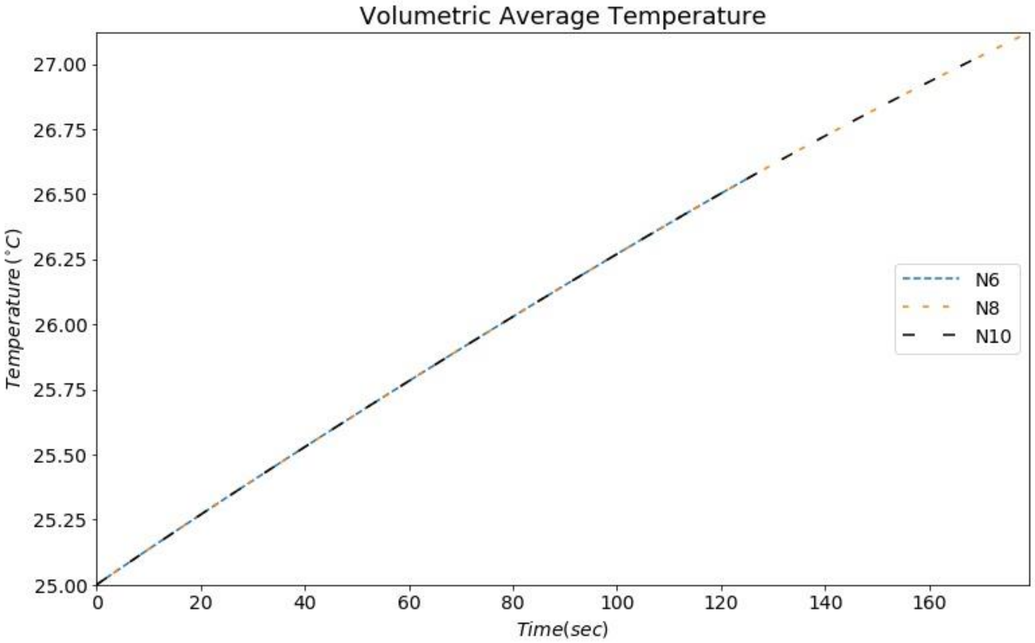


Figure 41: Volumetric Average Temperature Comparison

Represented in Figure 42, below, is a comparison of the pointwise difference in temperature values between each run. The magnitude of the largest temperature difference, w.r.t. polynomial order and common to each scenario, is $\sim 1.0 \times 10^{-5}$. As time increases, the lower order polynomial simulations tend to have a higher thermal value.

A second run was submitted for N=6 and again the job terminated prematurely at approximately the same time step. The divergence of the fifth order polynomial RMS values from that of the stable simulations coupled with the consistent early termination of run, confirms the hypothesis that the N=6 simulation has too coarse of a resolution for DNS.

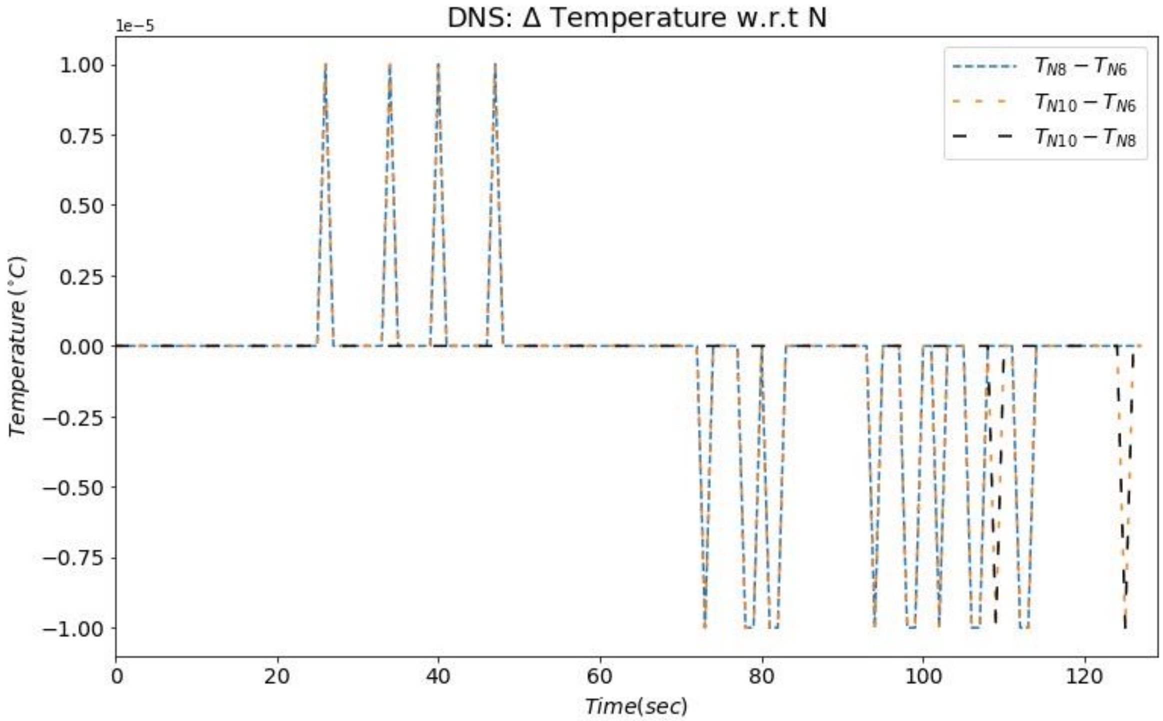


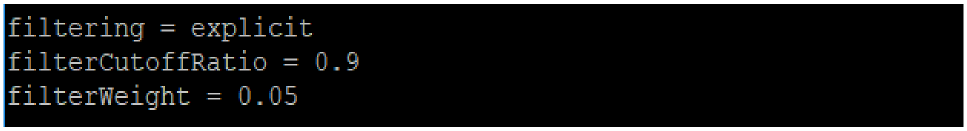
Figure 42: Comparison of Temperature Differences by Polynomial Order

6 Case 4: LES (NEK5000)

Following the analysis of several three-dimensional cases using DNS, the next goal of the study is to use NEK5000's Large Eddy Simulation (LES) method and determine the quasi-steady-state temperature.

6.1 Problem Definition

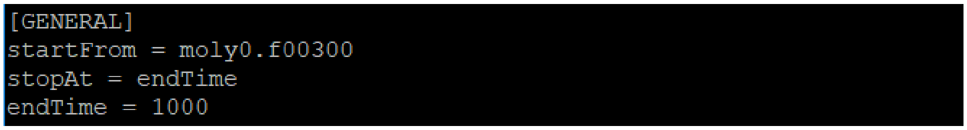
The three lines, displayed in Figure 43, are added to the [GENERAL] section of the *.par file for each case, shown in Appendix 1 below, reconfiguring them for the LES method. The values prescribed in these lines of code are a suggested starting point by the NEK5000 team at ANL. Unfortunately, documentation on LES settings for NEK5000 is limited and superficial making further understanding dependent on the user experience. The following evaluation is done using the suggested settings without expounding on the detail of their meaning, further investigation and experimentation is required to optimize the setup of the LES method using NEK5000.



```
filtering = explicit
filterCutoffRatio = 0.9
filterWeight = 0.05
```

Figure 43: moly.par LES Settings

In addition to the implementation of the LES method, simulation runtimes are extended using NEK5000's restart capabilities. The restart function is set by adding "startFrom" to the [GENERAL] section of the *.par file (Figure 44). "startFrom" is initialized to the desired restart output file/point, i.e. moly0.f00180. Use of the restart function here is possible by referencing the NEK5000's GitHub repository (Argonne National Laboratory 2018) example cases. Restarts were used to find both the simulation's quasi-steady-state temperature and the time at which it is reached.



```
[GENERAL]
startFrom = moly0.f00300
stopAt = endTime
endTime = 1000
```

Figure 44: moly.par Restart Setting

6.2 Simulation and Analysis

Outside of being able to run an N=6 simulation to later stages, transition from DNS to LES has little impact on solution accuracy. Figure 45 and Figure 46 plots the difference between LES and DNS temperature values with respect to polynomial order. Illustrated in Figure 45, when N=6, the DNS simulation clearly diverges during the evolution of the instabilities until failure time (t=127 seconds). Minimal variations between the methods occur for N=8 and the solutions are approximately identical for N=10.

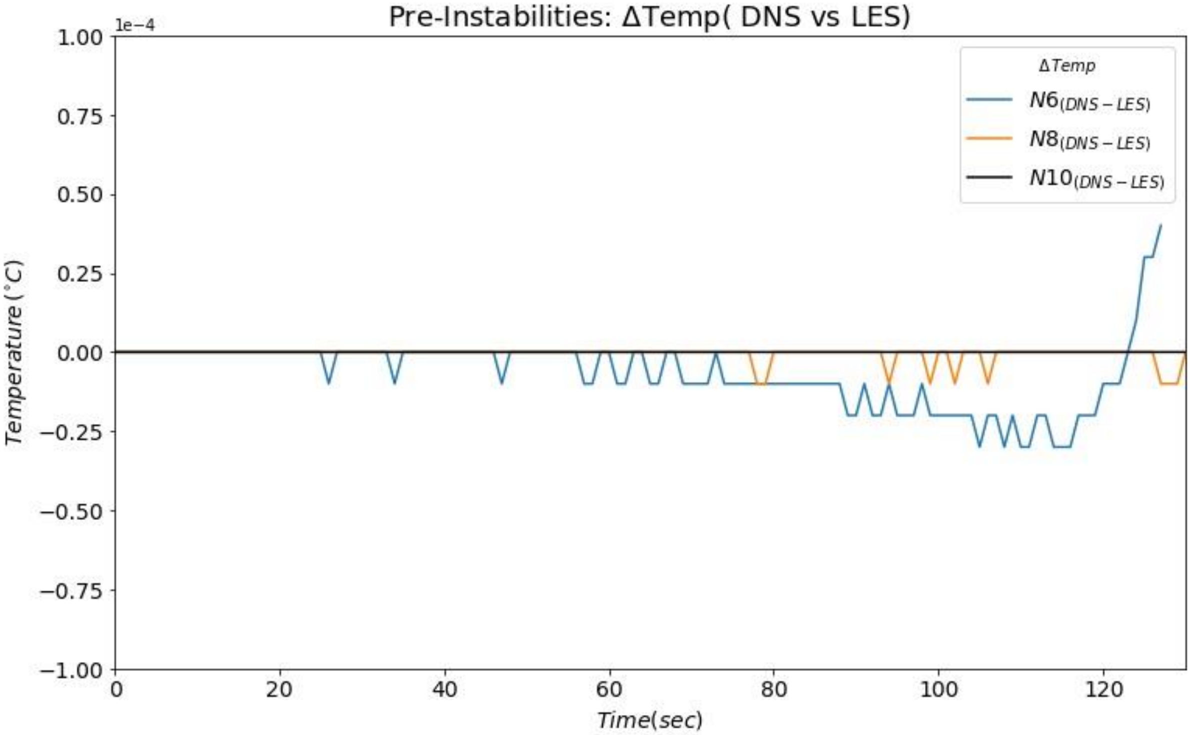


Figure 45: Temperature Difference (DNS - LES) Prior to Instabilities

The maximum deviation in temperature, between LES and DNS methods, is less than 0.01% (0.002 °C) of the initial temperature value for $0 \leq t < 180$. Figure 46 shows the temperature difference between the methods.

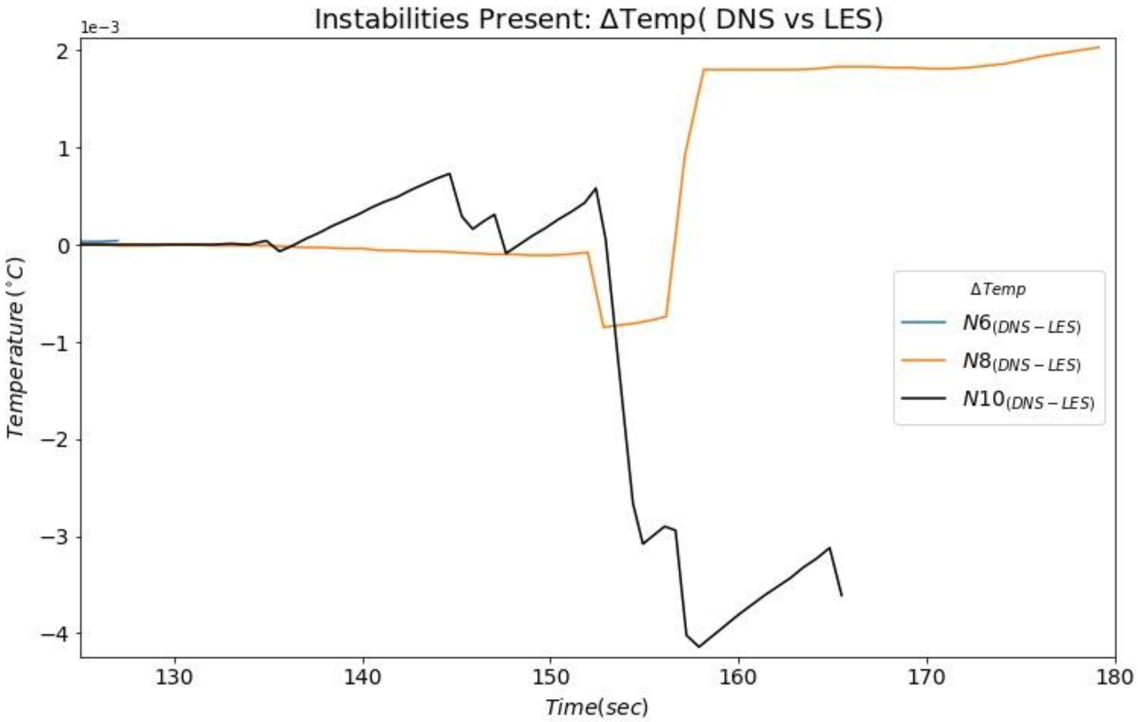


Figure 46: DNS-LES RMS Polynomial Order Comparison

While the DNS method diverged and terminated at t=127 when N=6, the LES method remained stable through every restarted simulation. Therefore, the LES method proved itself more robust than DNS, with respect to spectral element resolution.

6.2.1 Computation Time

The effect of using the LES solver on computational run time varies greatly depending on polynomial order. Compared to their respective DNS runs:

- N=6 results in a 14% slower runtime
- N=8 results in fractional decrease in runtime (<0.1%)
- N=10 results in a 19% faster runtime

Recalling from Section 5.2.2, setting $N = 6$ & 8 results in approximately 79% and 67% faster runtimes than when N=10, respectively. With the LES solver, $N = 6$ and 8 maintain significantly faster runtimes than N=10, 71% and 60% respectively, see Table 5. This is more meaningful now that analysis for all three polynomial orders is possible at later times, i.e. the transient through quasi-steady state regime. Computational time required, when N=6, to reach the quasi-steady states is approximately two weeks. From the reported statistics, we can extrapolate the respective computational runtime requirements for N= 8 & 10 to be approximately three and five weeks. Simulation runtime is also greatly dependent on the scaling of the simulations and as polynomial order increased so did processor requirements. For each of the simulations reported we are using 12 core intel processors. In the following section we discuss the results from the stable simulations of each polynomial order using the LES method.

Table 5: LES-Polynomial Order's Effect on Runtime at t =127 Seconds Simulation Time

N	6	8	10
Computational Runtime (Days)	1.541	2.172	5.3

6.2.2 Precision of Solution

As explained previously, the DNS solutions are relatively consistent for each run during the first few minutes (Figure 41). Recall that when N=6, the solution was shown to diverge prior to termination. The LES method was able resolve the physical processes of the least refined polynomial resolution, when the DNS could not. This is accomplished by computing the effect of the large scales directly (DNS) and modeling the small sub grid scales (SGS) (Germano, et al. 1991).

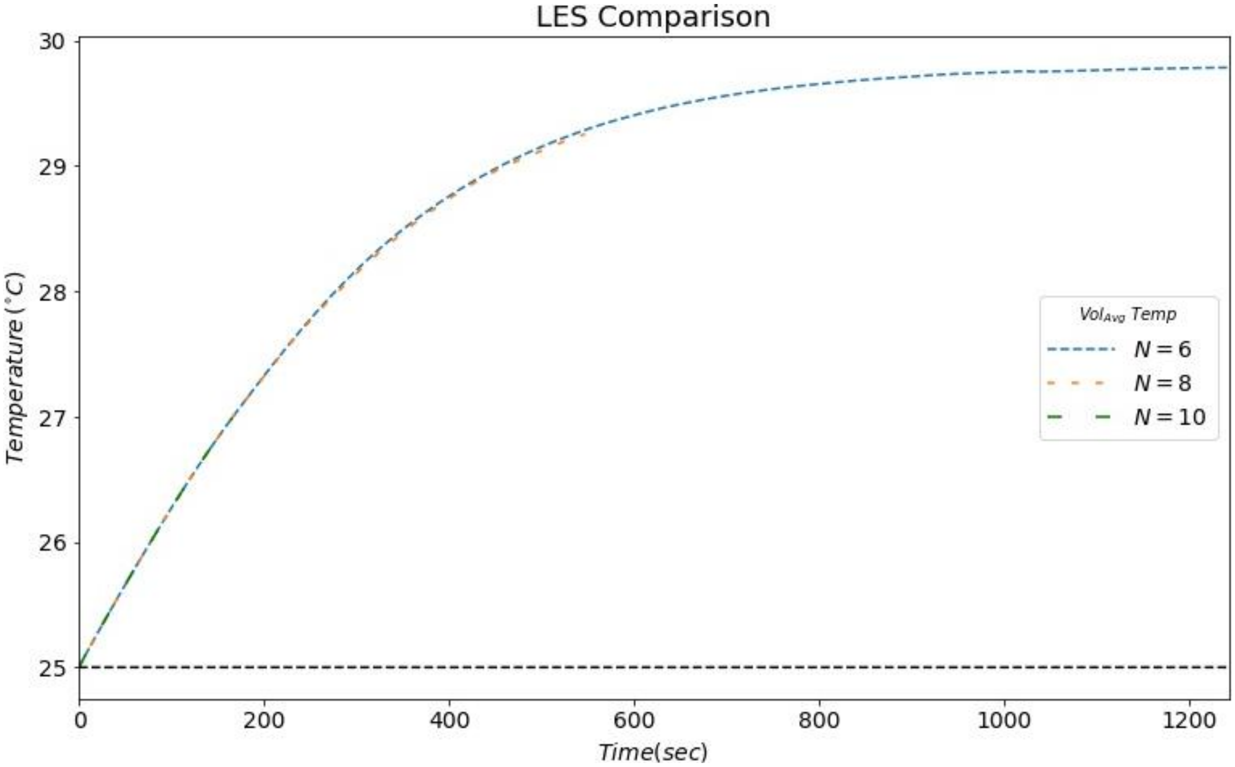


Figure 47: LES Volumetric Average Temperature Comparison

Approaching 4 minutes simulation time, when N=8, we see solution begin to slightly diverge from the N=6 solution. After approximately 7 ½ minutes simulation time the temperature stays begins to plateau between 29 and 30 degrees Celsius for both N=6 & N=8. The restart function enables us to determine the quasi-steady thermal state of 29.75 °C at approximately 15 minutes for N=6, see Figure 47.

7 Conclusion and Future Work

Users developing new simulations in NEK5000 must rely on the vast majority of their reference material to come from literature review, developer published documentation and examples of others previous work. While the aforementioned tools are helpful in case development, consistent documentation for end-users of Nek5000 is unfortunately lacking.

There are several key takeaways from the progression of case studies. The initial study, Case 1, showed how to implement thermal and velocity conditions in the various user files. Although each case contributed to this user's understanding on how and where to modify user files, the development of Case 2 proved the most instrumental in learning to implement changes in problem definition. Cases 1 & 2 reinforced the impact Reynolds and Rayleigh numbers has on flow. Transitioning from Case 2 to 3 introduced generating a computational domain in 3 dimensions using NEK5000 and Python. The third case study also introduced and affirmed the importance of the Boussinesq approximation when simulating buoyancy-driven convective flow.

Case 4, the culmination of the study, produced a quality CFD analysis including: a parametric study on model settings and detailed visualizations of the natural circulation phenomena occurring in a simplified AHR test condition. Increasing the polynomial order, i.e. spectral resolution, was proven to have the following impacts:

- Higher resolution visualizations, e.g. larger output files
- More stable simulations
- Greater computational cost, i.e. increase in processor requirements and runtimes

Therefore, trial runs are required to optimize available resources while maintaining stability for DNS runs. Stability was not an issue for the LES runs in NEK5000 in this study, however, more time is needed to evaluate and maximize usage of the LES solver. Since minimal documentation and references are available, w.r.t. the LES method, time and experience are required in order to optimize future simulations. A summary of the work done and key points of this study are as follows:

- DNS and LES methods were used to solve identical simulations in NEK5000
- A fixed mesh geometry of varying spectral resolution was analyzed for both methods
- For both methods, spectral resolution has minimal impact on solutions temperature
- Future simulations should begin being developed using the LES method with N=6

Using the prescribed test conditions of Case 4, a similar Finite Volume Method (FVM) Reynolds Averaged Navier-Stokes (RANS) simulation was run with commercial CFD tool "ANSYS-fluent, 2019R1". The volume averaged steady-state solution temperature was predicted to be 29.25 ℃, demonstrating good agreement with NEK5000's high resolution calculations. While the detailed turbulence and flow instabilities captured with NEK5000 are not observable with ANSYS-Fluent's calculations, the key system response quantities (SRQs), e.g. solution temperature and overall heat transfer coefficient, are reasonably well matched with the high-resolution simulations.

The next step involves using the large eddy simulation (LES) method to run an identical simulation using Fluent, thereby providing a benchmark study of NEK5000's capabilities.

8 References

Argonne National Laboratory. 2018. *Nek5000.* 12 19. Accessed 12 3, 2018. https://github.com/Nek5000.

—. 2018. *NEK5000.* July 24. Accessed June 4, 2018. https://nek5000.mcs.anl.gov/.

Birkhoff, Garrett. 1954. *Taylor Instability and Laminar Mixing.* Los Alamos: Los Alamos Scientific Laboratory.

Daly, Bart J. 1967. "Numerical Study of Two Fluid Rayleigh-Taylor Instability." *The Physics of Fluids* 297-307.

Deville, M. O., P. F. Fischer, and E. H. Mund. 2002. "Fluid Mechanics and Computation: An Introduction." In *Fluid Mechanics and Computation: An Introduction*, by M. O., P. F. Fischer, and E. H. Mund Deville, 1-32. Cambridge: Cambridge University Press.

Germano, Massimo, Ugo Piomelli, Parviz Moin, and William H. Cabot. 1991. "A Dynamic Subgrid-Scale Eddy Viscosity Model." *Physics of Fluids A: Fluid Dynamics 3* 1760-1765.

Henkes, R.A. W.M., and C. J. Hoogendoorn. 1990. "On the Stability of the Natural Convection Flow in a Square Cavity Heated from the Side." *Applied Scientific Research* 195-220.

Holman, Jack P. 2010. *Heat Transfer, 10th Edition.* McGraw-Hill Publishing Company.

Hussaini, M. Y., and T. A. Zang. 1986. *Spectral Methods in Fluid Dynamics.* NASA Contractractor 178103, Hampton, Virginia: NASA.

International Atomic Energy Agency. 2008. *HOMOGENEOUS AQUEOUS SOLUTION NUCLEAR REACTORS FOR THE PRODUCTION OF Mo-99 AND OTHER SHORT LIVED RADIOISOTOPES.* Vienna: International Atomic Energy Agency. 1-11.

Janssen, R. J. A., and R. A. W. M. Henkes. 1995. "Influence of Prandtl Number on Instability Mechanisms and Transition in a Differentially Heated Square Cavity." *Journal of Fluid Mechanics* Vol 290, pp. 319-344.

Julia C. Mullarney, Ross W. Griffiths and Graham O. Hughes. 2004. "Convection driven by differential heating at a horizontal boundary." *Journal of Fluid Mechanics* Vol. 516, pp. 181–209.

Kim, Seung Jun, and Cynthia Buechler. 2016. *Development of Multi-physics (Multiphase CFD + MCNP) Simulation for Generic Solution Vessel Power Calculation.* Milestone, Los Alamos: Los Alamos National Laboratory.

LeQuere, P. 1991. "Accurate Solutions to the Square Thermally Driven Cavity at High Rayleigh Number." *Computers & Fluids* Vol. 20, No. 1, pp 29-41.

Lindeburg, Michael R. 2016. *Mechanical Engineer Reference Manual for the PE Exam.* Belmont, CA: Professional Publications, Inc.

Mario R. Naranjo, R. Blake Wilkerson, Seung Jun Kim. 2018. *LA-UR-18-29657: The ARGUS Solution Reactor and Molybdenum Production:A Summary Report Based on Open Literature.* Summary Report, Los Alamos: Los Alamos National Laboratory.

Markatos, N. C., and K. A. Pericleous. 1984. "Laminar and Turbulent Natural Convection in an Enclosed Cavity." *International Journal of Heat and Mass Transfer* Vol. 27, No. 5, pp755-772.

MatWeb, LLC. 2018. *MatWeb.* July 24. Accessed July 01, 2018. http://www.matweb.com/search/DataSheet.aspx?MatGUID=64a0e072a9ec430e92ed984c7131b690&ckck=1.

Naranjo, Mario R. 2018. "gitlab.lanl.gov." *headNodeNekInstallFiles.* 12 17. Accessed 01 17, 2019. git@gitlab.lanl.gov:naranjom/headNodeNekInstallFiles.git.

Patera, Anthony T. 1984. "A Spectral Element Method for Fluid Dynamics: Laminar Flow in a Channel Expansion." *Journal of Computational Physics* 468-488.

Richard, Jacques C. 2019. *Spectral Element Method, Background and Details.* 02 01. Accessed 10 09, 2018. http://aeweb.tamu.edu/Richard/SEM.pdf.

Schlatter, Philipp. 2010. "Notes Spectral Methods." *Department of Mechanics, KTH.* 03 01. Accessed 10 11, 2018. https://www.mech.kth.se/~ardeshir/courses/literature/Notes_Spectral_Methods.pdf.

Sharp, D.H. 1984. "An Overview of Rayleigh-Taylor Instability." *Physica 12D* 3-18.

Thome, John Richard, and Filippo Cataldo. 2018. "ME341 - Heat and Mass Transfer." *EPFL WIKIS.* 05 30. Accessed 10 18, 2018. https://wiki.epfl.ch/me341-hmt/documents/lectures/slides_06_Laminar%20and%20turbulent%20boundary%20layers.pdf.

United States Nuclear Regulatory Commission. 2018. *Backgrounder on Research and Test Reactors.* June 06. Accessed January 18, 2019. https://www.nrc.gov/reading-rm/doc-collections/fact-sheets/research-reactors-bg.html#reactors.

Vujinovic, Aleksandar, and Joze Rakovec. 2015. "Seminarji." *University of Ljubljana Faculty of Mathematics and Physics*. 05 4. Accessed 02 21, 2019. http://mafija.fmf.uni-lj.si/seminar/files/2014_2015/KHI.pdf.

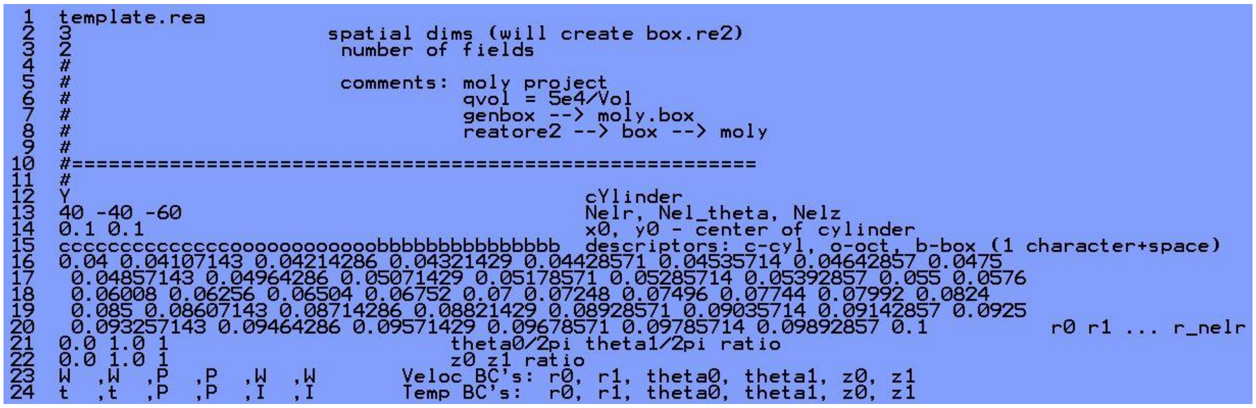
Appendices

1 Input Files

Parameters and conditions of the input files key to the development of Case 4 are briefly discussed in the following:

1.1 moly.box File

Defining Case 4's geometry and boundary conditions was the most difficult obstacle to overcome during this study. The domain is outlined using cylindrical coordinates, explicitly defining elemental geometry. This involved establishing where in the domain the center is located and where or how the element boundaries are determined. Radially the values began at cylinder wall and ended at a "box" wall. Azimuthally the domain is evenly divided between 0 to 2π. . The domain is also evenly divided axially over the range of the height. The following line numbers refer to Figure 48, a screenshot of the moly.box file. Line 1 refers to the "legacy" method mentioned above, here an example rea file, copied to the directory, is used as a template when the script genbox is called defining the geometry and its associated boundary conditions. Line 12 denotes that the geometry will be defined using cylindrical coordinates, first character (Y) is the only character used by the program. Line 13 defines 40 elements in the radial & azimuthal directions and 60 elements in the axial, the minus sign (-) in preceding the 60 and second 40 denote elements will be sized via line 21 & 22 inputs. Paradoxically, line 14 sets the center of the geometry at Cartesian coordinates (0.1, 0.1). Line 15 defines the shape of the element boundaries, 'c' denotes a circle, 'o' denotes an octagon and 'b' denotes a box. Line 16 defines the magnitude of the radial element's boundaries: beginning with the cylindrical cooling channel, diameter equal to 8 cm and ending with the domain length/width equal to 20 cm. Line 21 list the start and endpoint of the azimuthal elements divided by 2π, followed by the ratio the programmatically determined elements have to each other (1=equally divided elements). The last two lines define the boundary conditions for velocity and temperature. Dirichlet boundary conditions are: 'W' denoting the no-slip condition and 't' denoting constant temperature. The Neumann boundary condition is insulated 'I' (flux = 0). A new boundary condition applied is periodic, for both velocity and temperature. The periodic boundary condition is implemented in the theta (azimuthal) direction.



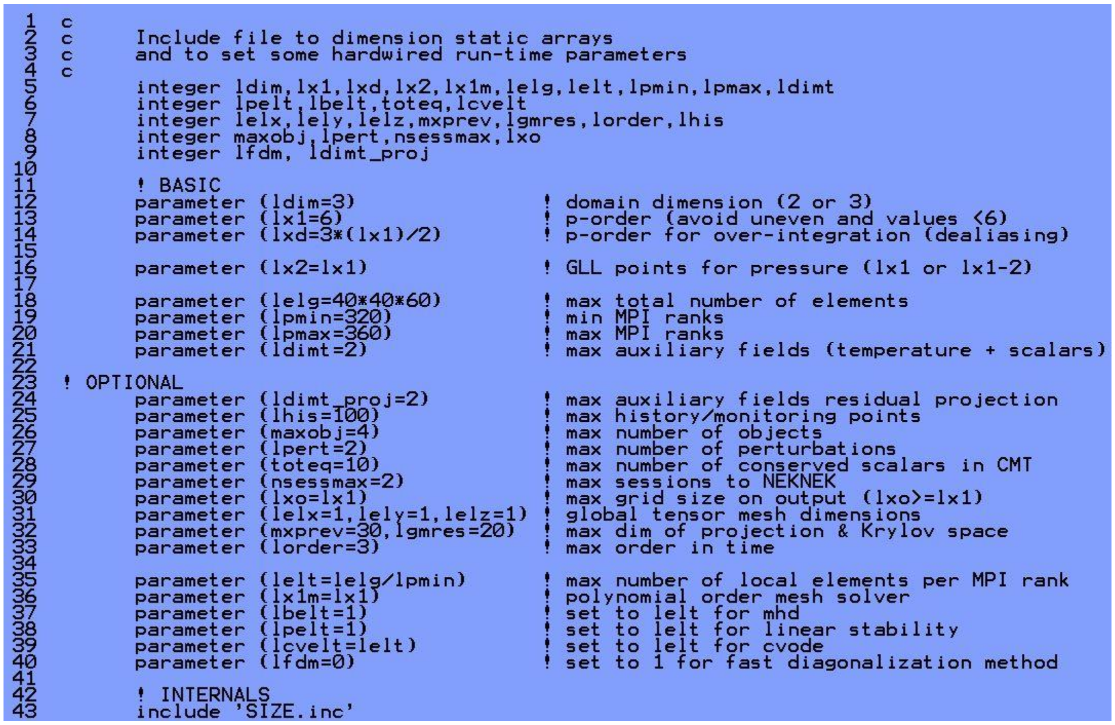
```
 1  template.rea
 2  3                          spatial dims (will create box.re2)
 3  2                           number of fields
 4  #
 5  #                           comments: moly project
 6  #                                     qvol = 5e4/Vol
 7  #                                     genbox --> moly.box
 8  #                                     reatore2 --> box --> moly
 9  #
10  #=====================================================
11  #
12  Y                                                 cYlinder
13  40 -40 -60                                        Nelr, Nel_theta, Nelz
14  0.1 0.1                                           x0, y0 - center of cylinder
15  ccccccccccccccoooooooooooobbbbbbbbbbbbbbb         descriptors: c-cyl, o-oct, b-box (1 character+space)
16  0.04 0.04107143 0.04214286 0.04321429 0.04428571 0.04535714 0.04642857 0.0475
17   0.04857143 0.04964286 0.05071429 0.05178571 0.05285714 0.05392857 0.055 0.0576
18   0.06008 0.06256 0.06504 0.06752 0.07 0.07248 0.07496 0.07744 0.07992 0.0824
19   0.085 0.08607143 0.08714286 0.08821429 0.08928571 0.09035714 0.09142857 0.0925
20   0.093257143 0.09464286 0.09571429 0.09678571 0.09785714 0.09892857 0.1          r0 r1 ... r_nelr
21  0.0 1.0 1                             theta0/2pi theta1/2pi ratio
22  0.0 1.0 1                             z0 z1 ratio
23  W  ,W  ,P  ,P  ,W  ,W        Veloc BC's: r0, r1, theta0, theta1, z0, z1
24  t  ,t  ,P  ,P  ,I  ,I        Temp BC's:  r0, r1, theta0, theta1, z0, z1
```

Figure 48: Geometery File 'moly.box'

1.2 SIZE File

During case development, files were created/amended in a specific order because of dependencies on one another. The first file created was the *.box file, the next amended was the SIZE file (Figure 49). Several values in the SIZE file are dependent on three lines in the *.box file, i.e. those defining spatial dimension, field and element quantity. Line 13 of the *.box file is necessary to determine the values of SIZE file parameter's 'lelg', 'lpmin' and 'lpmax'. The SIZE file is well documented, see (Argonne National Laboratory 2018). Key to the discussion on this case, the polynomial order ($n = 'lx1' - 1$) of the simulation is defined on line 13 of the SIZE file.

Note: as mesh resolution increases, so must the 'lpmin' and 'lpmax'.

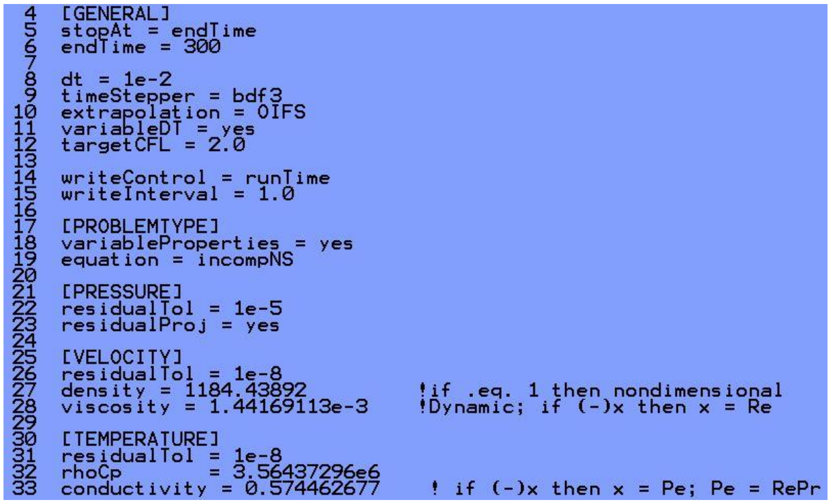


```
 1  c
 2  c     Include file to dimension static arrays
 3  c     and to set some hardwired run-time parameters
 4  c
 5        integer ldim,lx1,lxd,lx2,lx1m,lelg,lelt,lpmin,lpmax,ldimt
 6        integer lpelt,lbelt,toteq,lcvelt
 7        integer lelx,lely,lelz,mxprev,lgmres,lorder,lhis
 8        integer maxobj,lpert,nsessmax,lxo
 9        integer lfdm, ldimt_proj
10
11        ! BASIC
12        parameter (ldim=3)              ! domain dimension (2 or 3)
13        parameter (lx1=6)               ! p-order (avoid uneven and values <6)
14        parameter (lxd=3*(lx1)/2)       ! p-order for over-integration (dealiasing)
15
16        parameter (lx2=lx1)             ! GLL points for pressure (lx1 or lx1-2)
17
18        parameter (lelg=40*40*60)       ! max total number of elements
19        parameter (lpmin=320)           ! min MPI ranks
20        parameter (lpmax=360)           ! max MPI ranks
21        parameter (ldimt=2)             ! max auxiliary fields (temperature + scalars)
22
23  ! OPTIONAL
24        parameter (ldimt_proj=2)        ! max auxiliary fields residual projection
25        parameter (lhis=100)            ! max history/monitoring points
26        parameter (maxobj=4)            ! max number of objects
27        parameter (lpert=2)             ! max number of perturbations
28        parameter (toteq=10)            ! max number of conserved scalars in CMT
29        parameter (nsessmax=2)          ! max sessions to NEKNEK
30        parameter (lxo=lx1)             ! max grid size on output (lxo>=lx1)
31        parameter (lelx=1,lely=1,lelz=1) ! global tensor mesh dimensions
32        parameter (mxprev=30,lgmres=20) ! max dim of projection & Krylov space
33        parameter (lorder=3)            ! max order in time
34
35        parameter (lelt=lelg/lpmin)     ! max number of local elements per MPI rank
36        parameter (lx1m=lx1)            ! polynomial order mesh solver
37        parameter (lbelt=1)             ! set to lelt for mhd
38        parameter (lpelt=1)             ! set to lelt for linear stability
39        parameter (lcvelt=lelt)         ! set to lelt for cvode
40        parameter (lfdm=0)              ! set to 1 for fast diagonalization method
41
42        ! INTERNALS
43        include 'SIZE.inc'
```

Figure 49: SIZE File

1.3 moly.par File

Provided below in Figure 50 as a reference, is a screenshot of the parameter file for Case 4. Convergence of the solution was experimentally determined to be dependent upon the value of parameter 'dt': 'dt' determines the maximum time step value between its prescribed value and that calculated from the 'targetCFL'. The parameter file's options are well documented via the NEK5000 website (Argonne National Laboratory 2018).



```
[GENERAL]
stopAt = endTime
endTime = 300

dt = 1e-2
timeStepper = bdf3
extrapolation = OIFS
variableDT = yes
targetCFL = 2.0

writeControl = runTime
writeInterval = 1.0

[PROBLEMTYPE]
variableProperties = yes
equation = incompNS

[PRESSURE]
residualTol = 1e-5
residualProj = yes

[VELOCITY]
residualTol = 1e-8
density = 1184.43892          !if .eq. 1 then nondimensional
viscosity = 1.44169113e-3     !Dynamic; if (-)x then x = Re

[TEMPERATURE]
residualTol = 1e-8
rhoCp         = 3.56437296e6
conductivity = 0.574462677     ! if (-)x then x = Pe; Pe = RePr
```

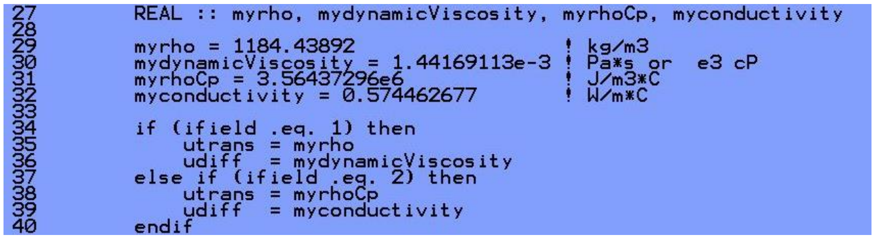
Figure 50: Parameter File

1.4 moly.usr File

Alongside the *.box file, the *.usr file required the most time understanding and is critical for any NEK5000 user to become familiar with it. The *.usr file enables one to control output, define BCs/ICs, modify mesh, etc. Written in Fortran 77 (F77), the *.usr file is a structured series of subroutines.

1.4.1 subroutine uservp

Shown in Figure 51 is an excerpt of the *.usr file's first subroutine 'uservp'. Recall from Appendix 1.1, two fields are defined. Per NEK5000 documentation, ifield = 1 and 2 refer to the momentum and energy equations, respectively. Following the logic for the order in which the files are created/amended, the same values for these four parameters are also initialized in Appendix 1.1.

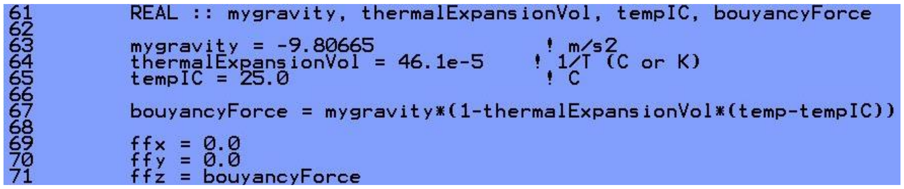


```
      REAL :: myrho, mydynamicViscosity, myrhoCp, myconductivity

      myrho = 1184.43892                    ! kg/m3
      mydynamicViscosity = 1.44169113e-3    ! Pa*s or  e3 cP
      myrhoCp = 3.56437296e6                ! J/m3*C
      myconductivity = 0.574462677          ! W/m*C

      if (ifield .eq. 1) then
          utrans = myrho
          udiff  = mydynamicViscosity
      else if (ifield .eq. 2) then
          utrans = myrhoCp
          udiff  = myconductivity
      endif
```

Figure 51: uservp Subroutine

1.4.2 subroutine userf

Figure 52 shows the implementation of the Boussinesq approximation, reference equation 28 above, using the 'userf' subroutine. Each NEK5000 prescribed term ffx, ffy and ffz, multiplied by density, is called once for every point: 'buoyancyforce' is defined as the Boussinesq approximation, less density, in the correct gravitational orientation (ffz).

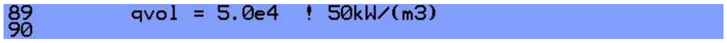


```
      REAL :: mygravity, thermalExpansionVol, tempIC, bouyancyForce

      mygravity = -9.80665                ! m/s2
      thermalExpansionVol = 46.1e-5     ! 1/T (C or K)
      tempIC = 25.0                       ! C

      bouyancyForce = mygravity*(1-thermalExpansionVol*(temp-tempIC))

      ffx = 0.0
      ffy = 0.0
      ffz = bouyancyForce
```

Figure 52: userf Subroutine

1.4.3 subroutine userq

To apply the volumetric heat generation that occurs by irradiating uranyl sulfate solution, q_{vol} from the NS energy equation 25 above is initialized to a predefined value (Figure 53).

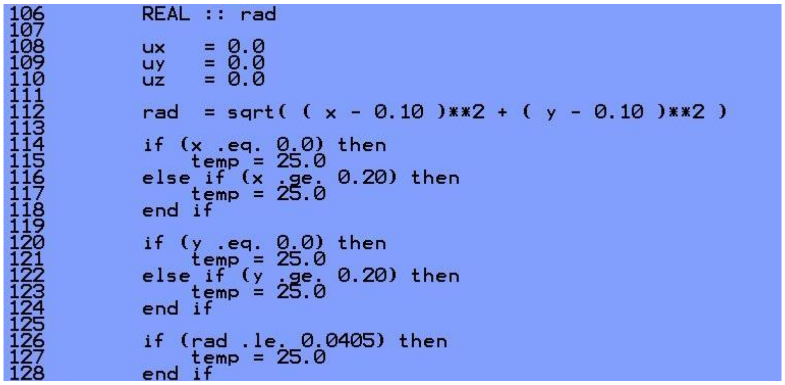


```
        qvol = 5.0e4  ! 50kW/(m3)

```

Figure 53: userq Subroutine

1.4.4 subroutine userbc

The subroutine 'userbc' assigns values to the boundary conditions applied in Appendix 1.1. The most challenging aspect of assigning values was determining how to apply them to the cooling channel wall, reference lines 112, 126, 127 and 128 in Figure 54 This challenge proposed a question: given that the temperature boundary conditions are applied in the *.box file, reference Appendix 1.1, why are these values not applied at the boundaries by simply initializing the temp variable? Perhaps another method exists that answers this question; the method illustrated below achieves the goal of applying constant temperature at the necessary points.

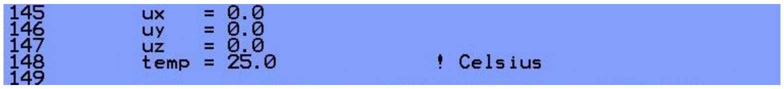


```
106        REAL :: rad
107
108        ux   = 0.0
109        uy   = 0.0
110        uz   = 0.0
111
112        rad  = sqrt( ( x - 0.10 )**2 + ( y - 0.10 )**2 )
113
114        if (x .eq. 0.0) then
115            temp = 25.0
116        else if (x .ge. 0.20) then
117            temp = 25.0
118        end if
119
120        if (y .eq. 0.0) then
121            temp = 25.0
122        else if (y .ge. 0.20) then
123            temp = 25.0
124        end if
125
126        if (rad .le. 0.0405) then
127            temp = 25.0
128        end if
```

Figure 54: userbc Subroutine

1.4.5 subroutine useric

Subroutine useric works similar to userbc applying velocity and temperature values to computational domain points.



```
        ux   = 0.0
        uy   = 0.0
        uz   = 0.0
        temp = 25.0             ! Celsius

```

Figure 55: useric Subroutine